

콜티 (Call Me Trash To Treasure)



ESG 실현하는
호출 가능한 인공지능 쓰레기통 로봇
“Call T”

호예가(호기심 많은 예술가)
2104676 이지현
2104607 김서영
2105707 박소영
2104705 최보현

ESG 실현하는 호출 가능한 인공지능 쓰레기통 로봇 “콜티(Call Me Trash To Treasure)”

본 제품은 ESG 원칙에 중점을 두어 쓰레기 무단투기 방지와 효율적인 쓰레기 관리로 지속 가능한 환경을 조성하고, 기업의 사회적 책임을 수행할 수 있습니다. 또한, 환경적 영향을 최소화하고, 재활용을 촉진하여 자원의 효율적인 이용을 추구함으로써 ESG 경영에 기여할 수 있습니다. 기업 뿐 아니라 일반 대중들에게도 분리수거 관련 지식을 제공하여 환경 인식을 높이는 데 중점을 두기 때문에 사회적 측면에도 긍정적인 영향을 미칩니다. 저희는 페스티벌 현장부터 길거리까지 다양한 환경에서 쓰레기 관리의 혁신적인 해결책으로서 ESG 실천을 장려하고자 하여 이 제품을 개발하였습니다.

01

개요

- 팀원 간 모듈 분배
- 개발 배경
- SWOT 분석
- 벤치마킹
- 차별성
- 시장/기술 동향 분석

02

주요 내용

- 흐름도
- 유스케이스
- 알고리즘 명세서
- 개발환경 및 설명
- 요구사항 정의서
- 사용자 페이지
- 관리자 페이지
- 하드웨어/센서 회로도
- S/W 주요 기능
- H/W 주요기능 및 부품
- 기술 구현 프로세스
- 프로젝트 관리

03

기대효과

- 창출하고자 하는 사회적 가치

팀원 간 모듈 분배

이지현

Hardware & Software

- 웹사이트 개발
- 라즈베리파이 ↔ 아두이노 API 개발
- 로봇 제어 및 자율주행
- Image Deep Learning
- 주파수 및 신호 세기 측정

김서영

Hardware

- 로봇 외형 디자인
- 센서 제어 알고리즘
- 로봇 제어 및 자율주행

박소영

Hardware

- 로봇 외형 디자인
- 쓰레기 압축 및 웹 서버와의 통신

최보현

Sftware

- Image Deep Learning
- 웹 사이트 ↔ 아두이노 API 개발
- 주파수 및 세기 측정

개발 배경



페스티벌 현장에서의 쓰레기 다량 발생, 쓰레기 무단투기 급증

- 축제가 끝난 자리에 남아있는 쓰레기 산
- 쓰레기통이 멀다는 이유로 아무 곳이나 무단으로 쓰레기를 투기하는 행위 급증

쓰레기 분리수거에 대한 부족한 지식

- 분리수거에 대한 낮은 이해도
- 쓰레기 분리수거에 대한 안일한 생각을 가지고 있는 사람들

번거로운 쓰레기통 관리 문제

- 쓰레기통이 가득 차서 쓰레기통 주변에 쓰레기를 버리는 현상을 쉽게 찾아볼 수 있음.
- 수시로 쓰레기통을 관리해야 하는 번거로움을 해결하고자 함.

비용 절감과 효율성 증대

- 쓰레기 처리 및 재활용 비용을 절감하고, 쓰레기 관리 프로세스를 효율적으로 처리할 수 있는 제품을 개발하고자 함.

ESG STRATEGY

-
1. 무단 투기 예방
 2. 분리수거 교육 및 인식 강화
 3. 청정 페스티벌 도모

쓰레기
무단 투기

분리수거
문제

1. SWOT 분석

↳ S (Strength)

- 쓰레기통 호출 기능으로 사용자 편의성 증대
- 분리수거 문제해결 도움

▢ W (Weakness)

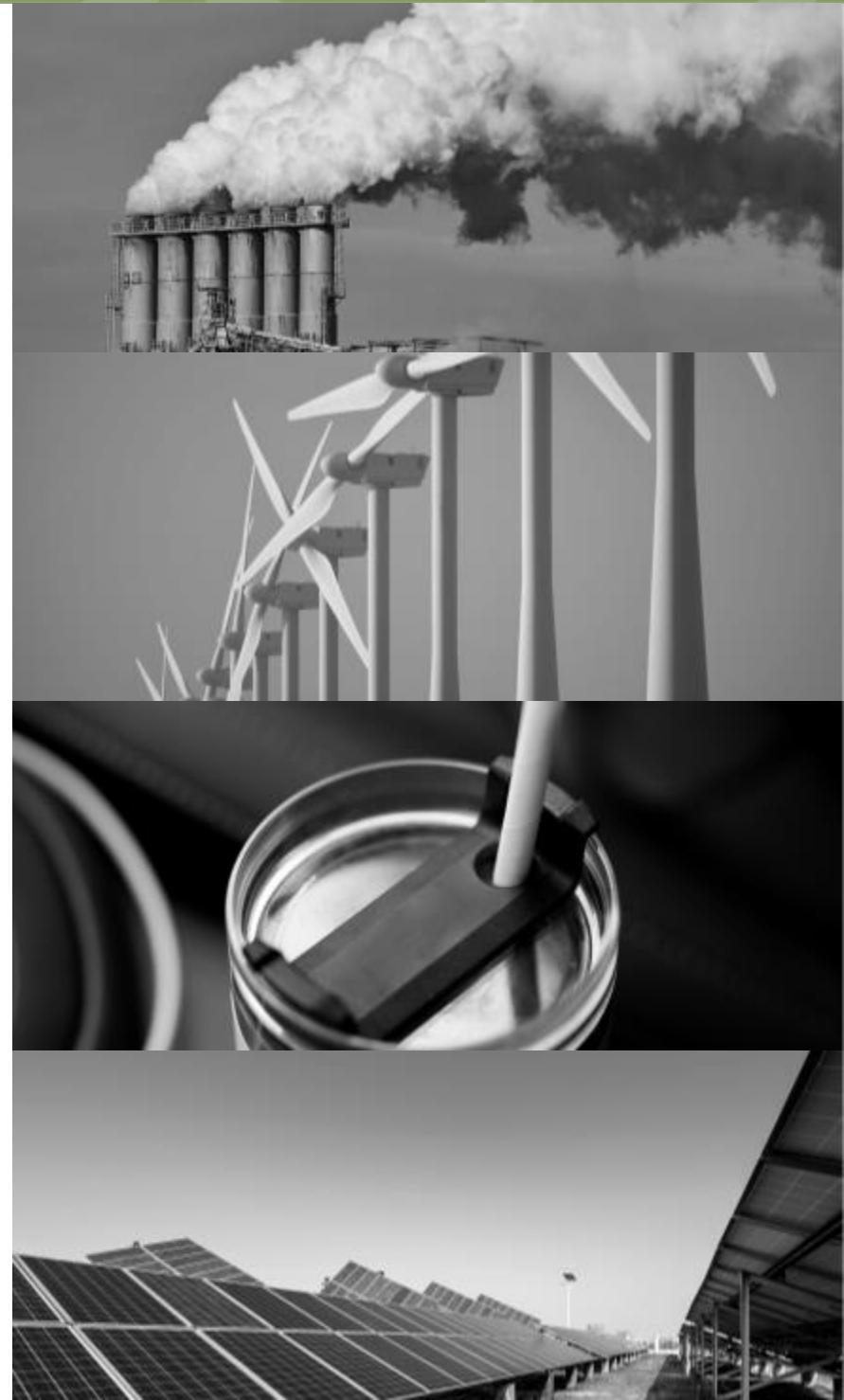
- 로봇 개발 시 많은 초기 비용 요구

▢ O (Opportunity)

- 매년 증가하는 쓰레기량과 환경 정책의 강화로 인해 쓰레기 처리 시장 성장
- 자율주행 로봇의 상용화

▢ T (Threat)

- 유사한 제품의 많은 출시
- 인공지능 로봇에 대한 낮은 이해도



1. 벤치마킹



네프론

AI 재활용 쓰레기통

- 센서 및 이미지 인식 기술 활용
- 잘못된 분리수거 방지
- 재활용률 높임



라티

자율주행 쓰레기통 로봇

- 적재량 확인 후 알림 표시
- 배출 위치로 이동
- 배터리 부족 시 충전 장소로 이동



클린큐브

태양광 압축 쓰레기통

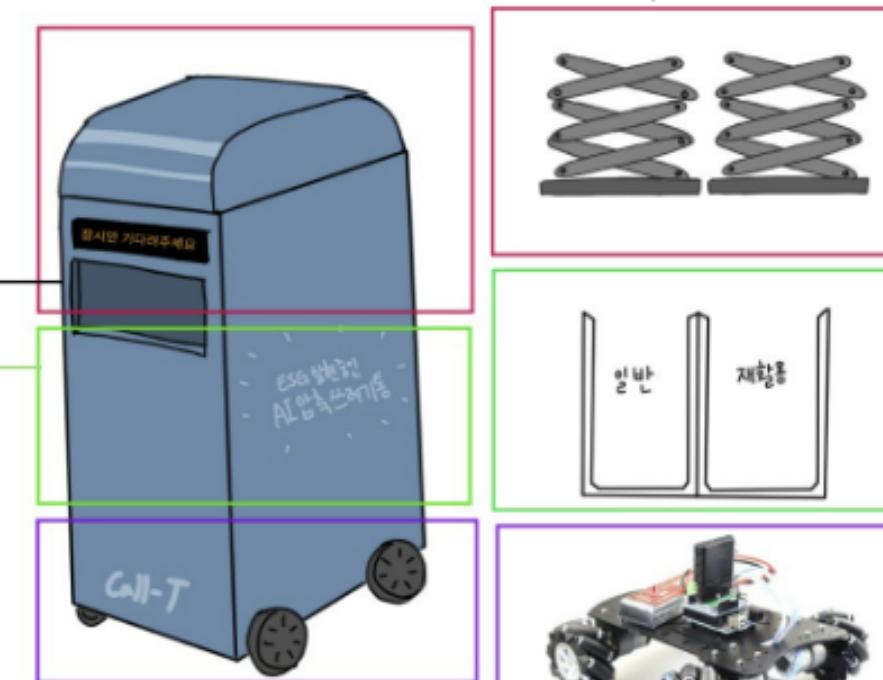
- 적재량 실시간 모니터링
- 최적의 수거 경로 계획
- 압축 기능으로 용량 최대 활용

1. 벤치마킹



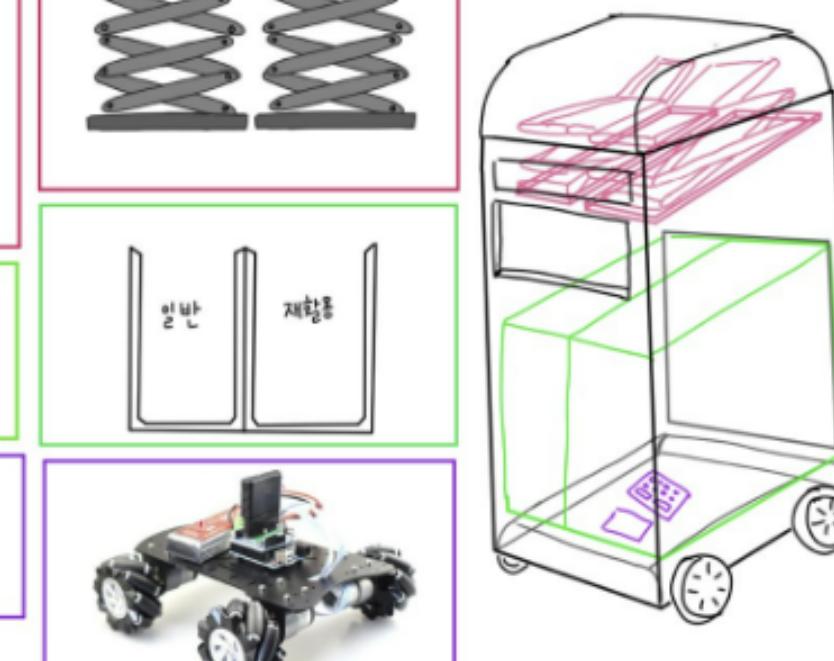
자동개폐 기능

가까이 있는 사람을 인식하면
자동으로쓰레기통 뚜껑 개폐



재활용 분류 기능

재활용 여부에 따라 쓰레기 분류



압축 기능(Cross Compactor)

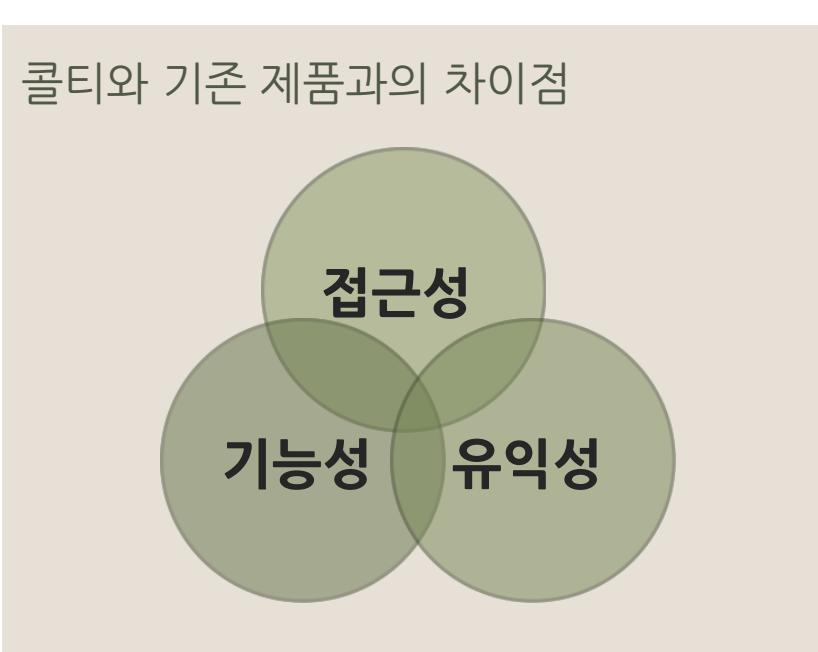
쓰레기 적재량이 일정 수준 이상이면
쓰레기통 자동으로 압축하여 수거 공간을
넓힘

자율주행(RC 메카님 훈)

호출자에게 장애물을 회피하며 이동

차별성

콜티만의 차별성



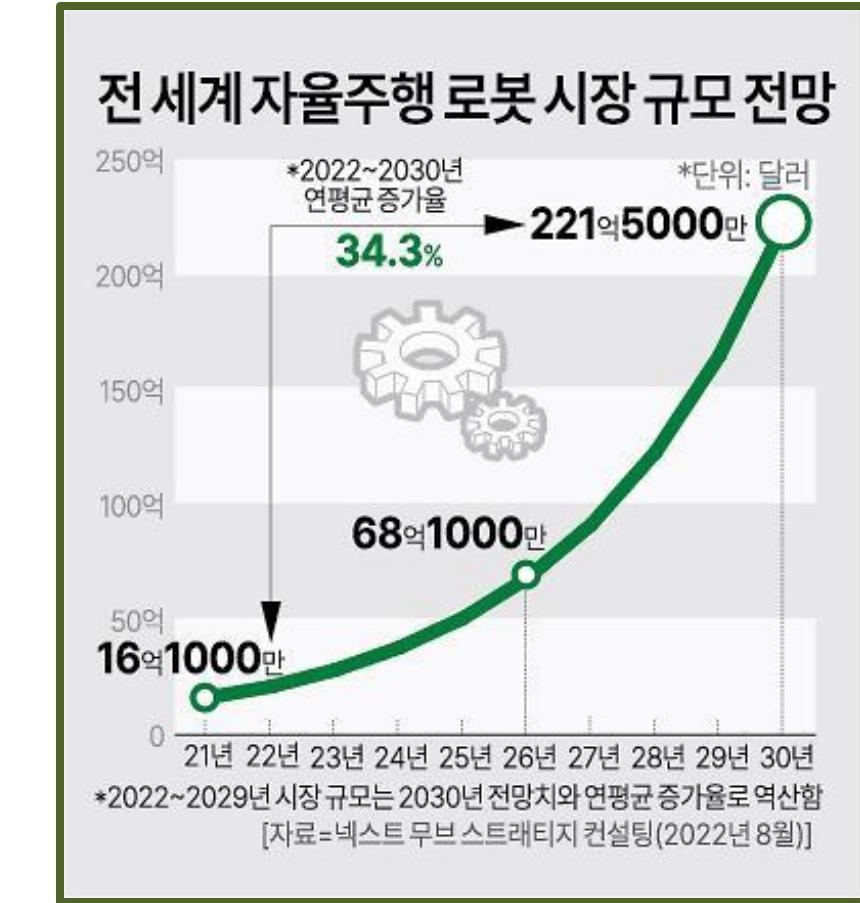
기능 비교표

업체명	콜티	네프론	라티	클린큐브
자율주행	O	X	O	X
호출기능	O	X	X	X
분리수거 기능	O	O	X	X
압축 기능	O	X	X	O
교육용 컨텐츠 제공	O	X	X	X

기존 제품과의 비교

측정지표	콜티	기존 쓰레기통 로봇
접근성	사용자가 쓰레기통을 찾아가지 않아도 호출하여 부를 수 있음	지정된 위치에만 존재하거나, 사용자 호출 여부에 관계없이 이동
호출기능	수거한 쓰레기를 분리수거 가능 여부에 따라 자동으로 1차 분류 진행	다양한 종류의 쓰레기를 하나의 쓰레기통에 수거
분리수거 기능	분리수거와 관련된 퀴즈 컨텐츠를 제공함으로써 분리수거에 대한 지식 함양 가능	단순히 시각적인 홍보용 정보만을 제공

시장/기술 동향 분석



자율주행 로봇의 시장 동향 분석

- 자율주행 로봇 시장은 코로나19에 따른 비대면 소비확산 추세, 자율주행 기술의 발전 등 다양한 요인으로 2030년까지 연평균 34.3% 성장세를 보일 것으로 전망됨
- 최근 개정된 ‘지능형 로봇 개발·보급 촉진법(이하 지능형로봇법)’이 연내에 시행되면 자율주행 로봇이 보도를 다니고 공원 등에 출입할 예정
- 이 분야 기술 개발에 투자해 온 국내 주요 정보통신기술(ICT) 업체들이 앞다퉈 서비스 상용화에 뛰어들 것으로 보임
→ 실외 자율주행 로봇이 상용화됨을 기대할 수 있음

흐름도



사용자

- ① 호출 버튼을 눌러 쓰레기통 호출
- ②-① 대기 시간동안 퀴즈 콘텐츠 이용 (선택)
- ②-② 쓰레기 투입



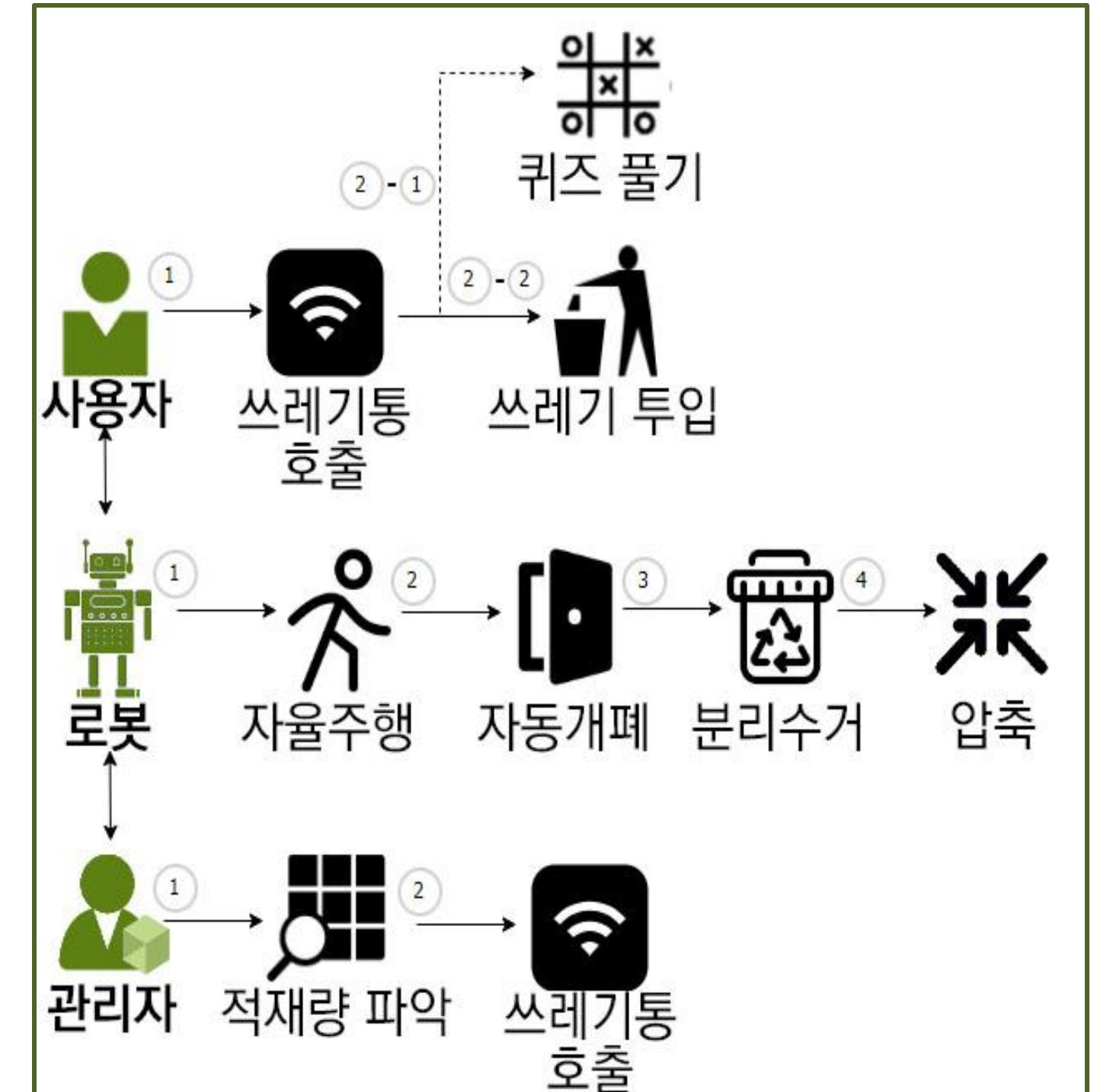
로봇

- ① 호출한 사용자의 위치를 파악해 이동
- ② 근접해 있는 사용자를 인식 후 입구 자동 개방
- ③ 보유중인 데이터를 활용해 쓰레기 분류 진행
- ④ 쓰레기가 일정량에 도달하면 자동으로 압축 진행

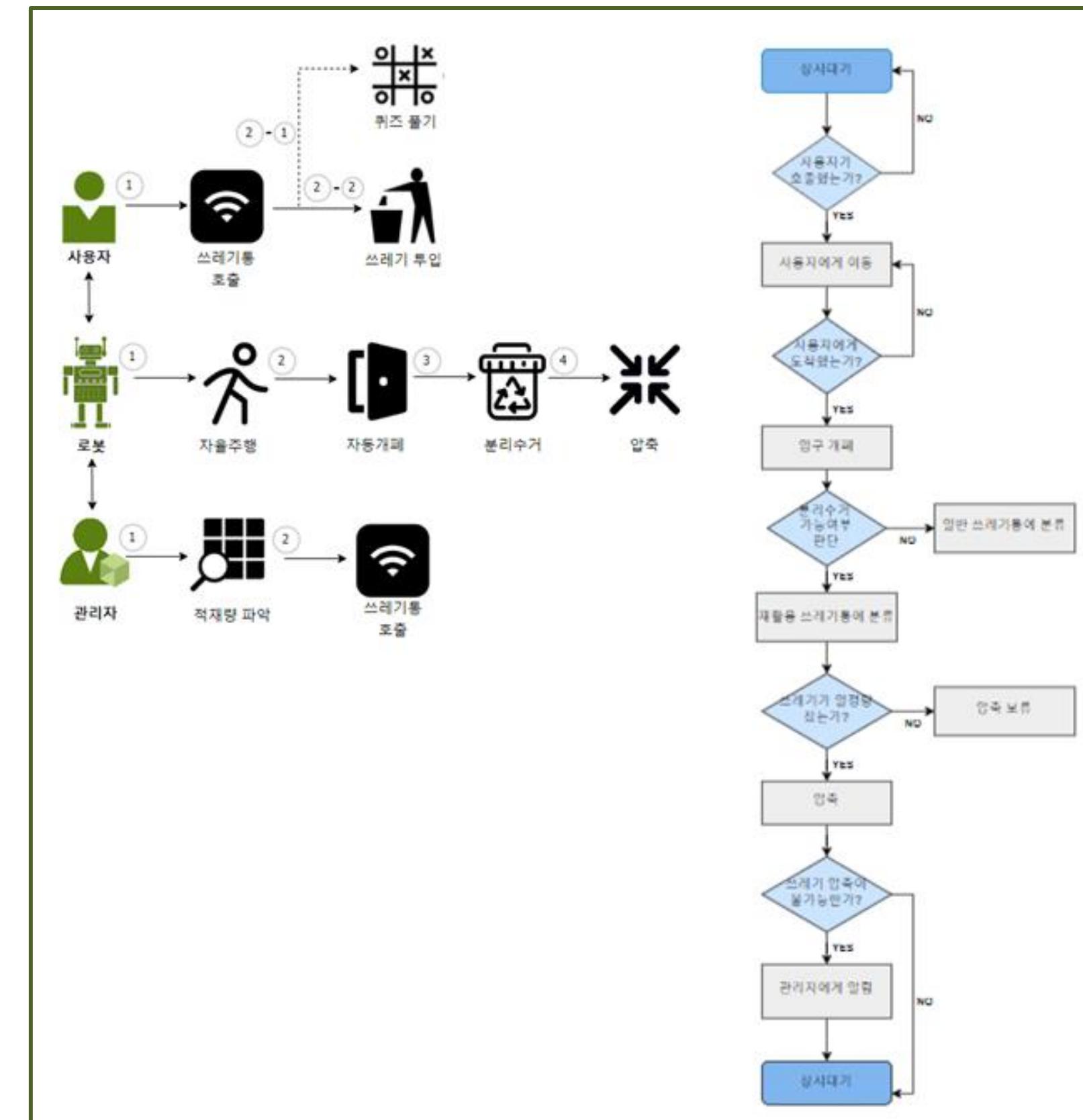


관리자

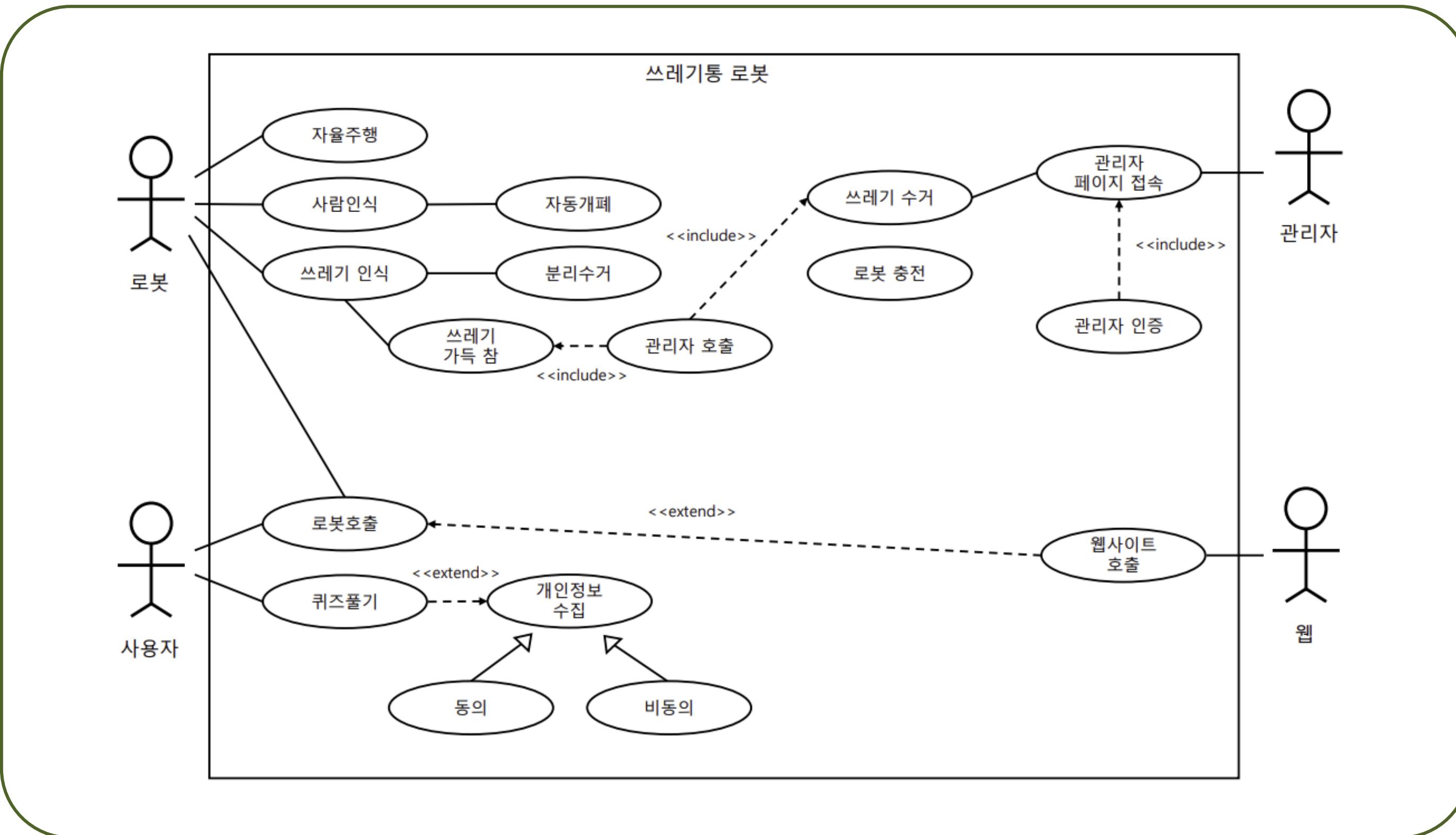
- ① 관리자 페이지를 통해 쓰레기 적재량 파악
- ② 압축이 불가할 경우 쓰레기통을 호출해 쓰레기 수거



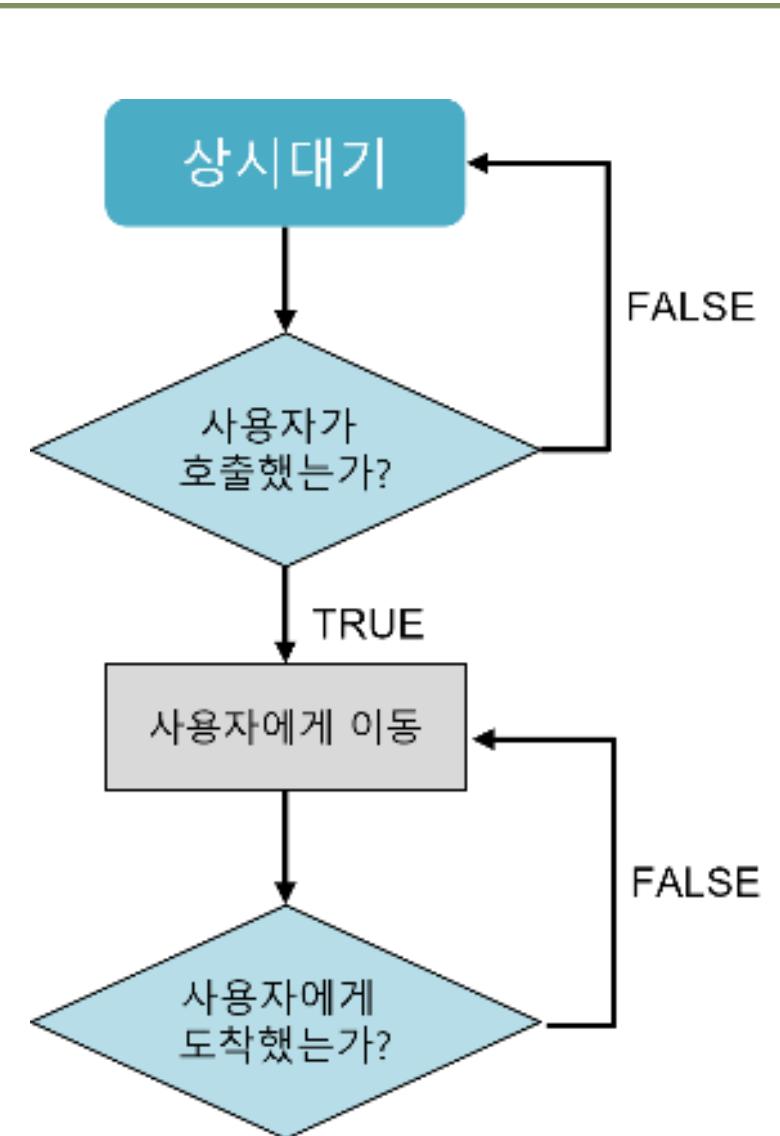
서비스 흐름도



유스케이스(Use case)



알고리즘 명세서



◆ 알고리즘 시나리오

➤ 사용자를 위한 기능

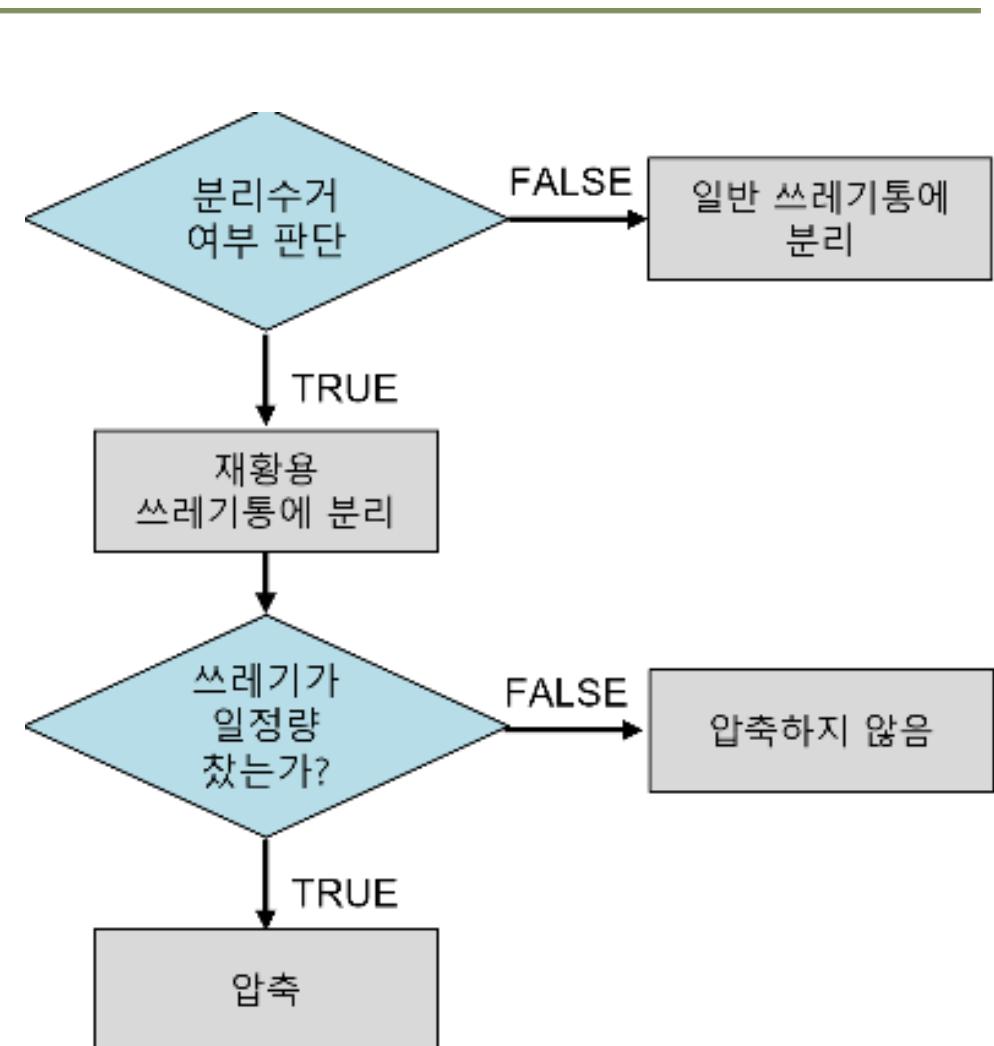
1. 상시 대기

2. 사용자가 호출 버튼으로 쓰레기통을 호출하면

RSSI값을 파악해 사용자에게로 자율주행

3. 장착된 초음파 센서와 LIDAR 센서를 사용해 장애물을
인식 후 회피하며 사용자에게 이동

알고리즘 명세서

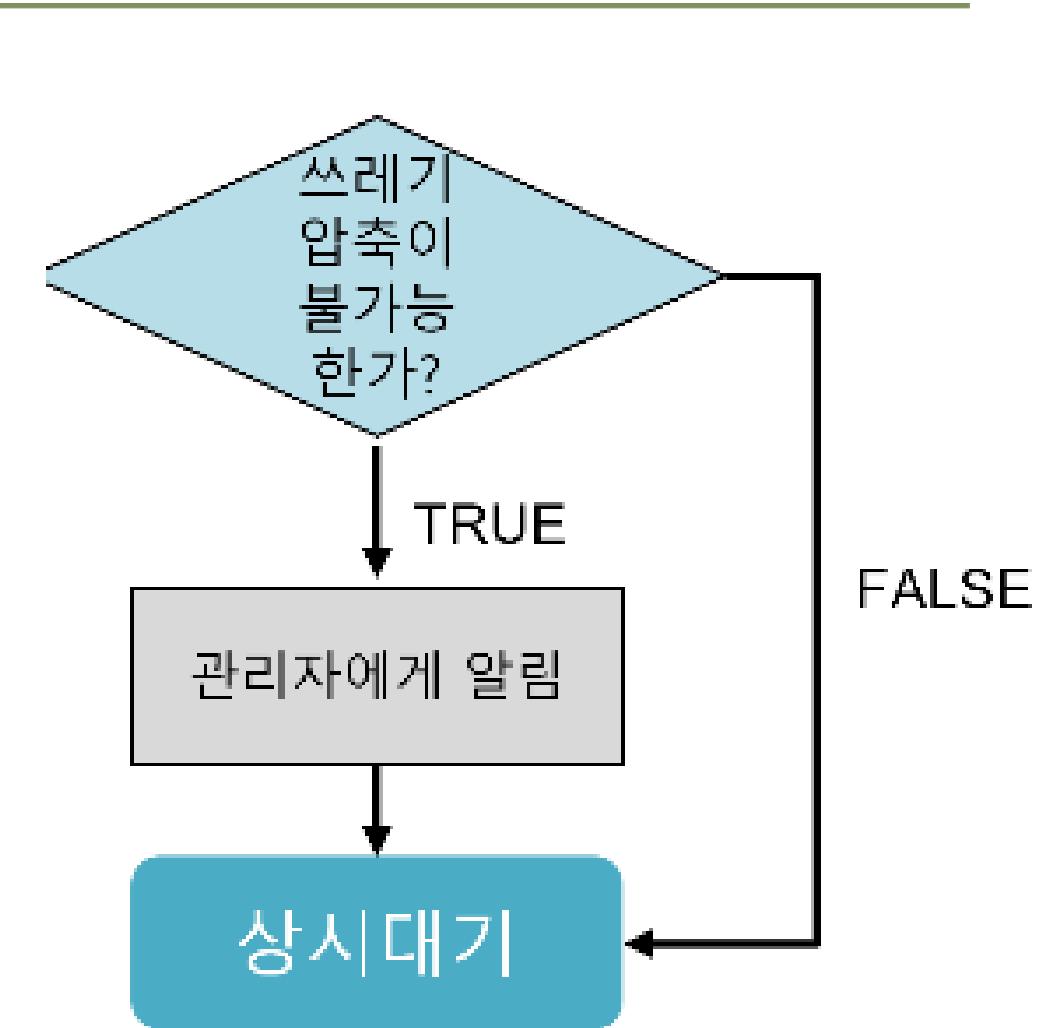


◆ 알고리즘 시나리오

➤ 쓰레기통 기능

1. 사용자로부터 쓰레기 수거
2. 커스텀 데이터셋으로 훈련된 YOLOv5 모델을 사용하여 쓰레기를 인식 후, 재활용 가능 여부에 따라 일반 쓰레기와 재활용 쓰레기로 분류
3. 쓰레기를 수거하는동안 내부에 장착된 초음파 센서를 통해 쓰레기 적재량을 실시간으로 파악
4. 쓰레기가 일정량 이상 가득 차면 리니어를 구동시켜 압축 진행

알고리즘 명세서



◆ 알고리즘 시나리오

➤ 관리자를 위한 기능

1. 쓰레기 적재량을 초음파 센서를 통해 **파악**
2. 설정한 일정량의 수준을 초과하면 **압축이 불가능하다고 판단**
- 3-1. 압축이 불가할 경우 **관리자에게 알림 기능 수행**
- 3-2. 일정량을 초과하지 않을 경우 상시 대기 상태 유지

개발 환경 및 설명

구분		상세내용
S/W 개발환경	OS	Windows 10, Linux, Ubuntu
	개발환경(IDE)	Arduino IDE, Visual Studio Code
	개발도구	ROS, React, openCV, firebase
	개발언어	Python, HTML, CSS, Javascript
	기타사항	NAVER Cloud
H/W 구성장비	디바이스	Arduino Mega 2560, RaspberryPi4, 모터 드라이버 보드
	센서	LIDAR(IP65), 초음파 센서(HC-SR04)
	통신	RSSI, serial, BLE, MQTT
	언어	Python, C/C++
	기타사항	Inventor(3D 모델링 도구)
프로젝트 관리환경	형상관리	GitLab
	의사소통관리	Notion, KaKao Talk, Zoom, 오프라인 회의
	기타사항	-

요구사항 정의서

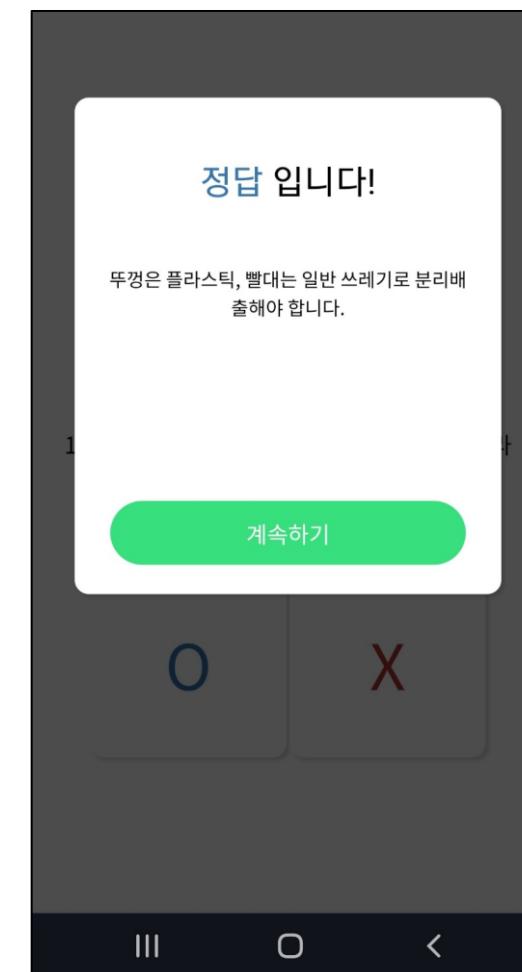
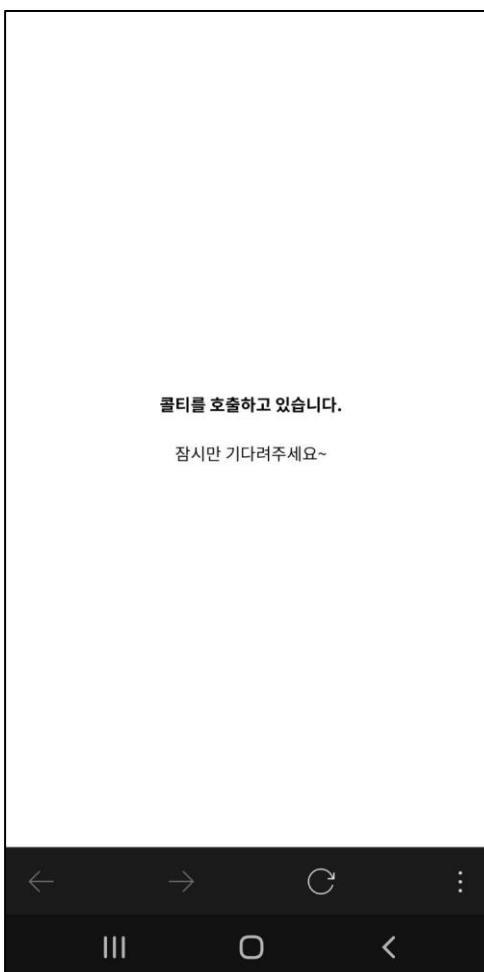
10

구분	기능	설명	구분	기능	설명
S/W	호출 기능	쓰레기통 로봇 '콜티'를 호출할 수 있는 기능 (사용자는 제공받은 URL 링크를 통해 메인 페이지 접속)	H/W	자율주행	호출한 사용자에게 장애물을 피해가며 자율주행으로 이동
	퀴즈 콘텐츠	쓰레기통을 기다리는 동안 분리수거 관련 지식을 향상시킬 수 있는 퀴즈 콘텐츠 제공		쓰레기 종류 파악	cam을 통해 투입된 쓰레기의 종류를 파악
	적재량 정보 확인	관리자가 실시간으로 쓰레기통의 적재량을 파악할 수 있도록 적재량에 대한 정보를 제공		분리수거 진행	파악한 쓰레기 종류를 분리수거 가능 여부에 따라 쓰레기통 내부에서 1차 분류 진행
	알림 기능	압축이 불가할 정도로 쓰레기통이 가득 찰 시 관리자에게 쓰레기 처리를 위한 알림 제공		압축	일정량 이상으로 쓰레기가 차면 압축기를 가동 시켜 쓰레기통 내부 공간 확보
				웹 서버 연결	웹 서버와 연동하여 통신이 가능하고, 각 센서들로부터 측정값을 제공받음

알고리즘 명세서

구분	요구사항ID	요구사항명	기능ID	기능명	상세 설명	필수 데이터	선택 데이터
사용자	A01	메인 페이지	A01_B01	호출하기	- 쓰레기통 로봇 '콜티'를 호출할 수 있는 버튼 (사용자는 제공받은 URL 링크를 통해 메인 페이지 접속)	위치 수집 약관 동의	
			A01_B02	퀴즈 풀기	- 분리수거 관련 퀴즈를 풀 수 있는 페이지로 넘어가는 버튼		
관리자	A02	로그인	A02_B01	일반 로그인	- 아이디 혹은 이메일, 비밀번호로 로그인할 수 있는 기능	ID 혹은 email, 비밀번호	
			A02_B02	소셜 로그인	- 아이디, 비밀번호 입력 없이 [구글], [깃허브] 소셜 수단으로 간편하게 로그인할 수 있는 기능		
관리자	A03	회원가입	A03_B01	일반 회원가입	- 아이디 혹은 이메일, 비밀번호로 회원가입하는 기능	ID 혹은 email, 이름, 비밀번호	
			A03_B02	소셜 회원가입	- 아이디, 비밀번호 입력 없이 [구글], [깃허브] 소셜 수단으로 간편하게 회원가입 할 수 있는 기능		
관리자	A04	관리자 페이지	A04_B01	홈 화면	- 관리자 간의 현재 쓰레기통 상태를 실시간으로 공유할 수 있는 페이지		사진 이미지
			A04_B02	적재량 확인	- 현재 쓰레기 적재량을 실시간으로 파악 할 수 있는 페이지		

사용자 페이지



메인 화면

호출 화면

단계 선택 화면

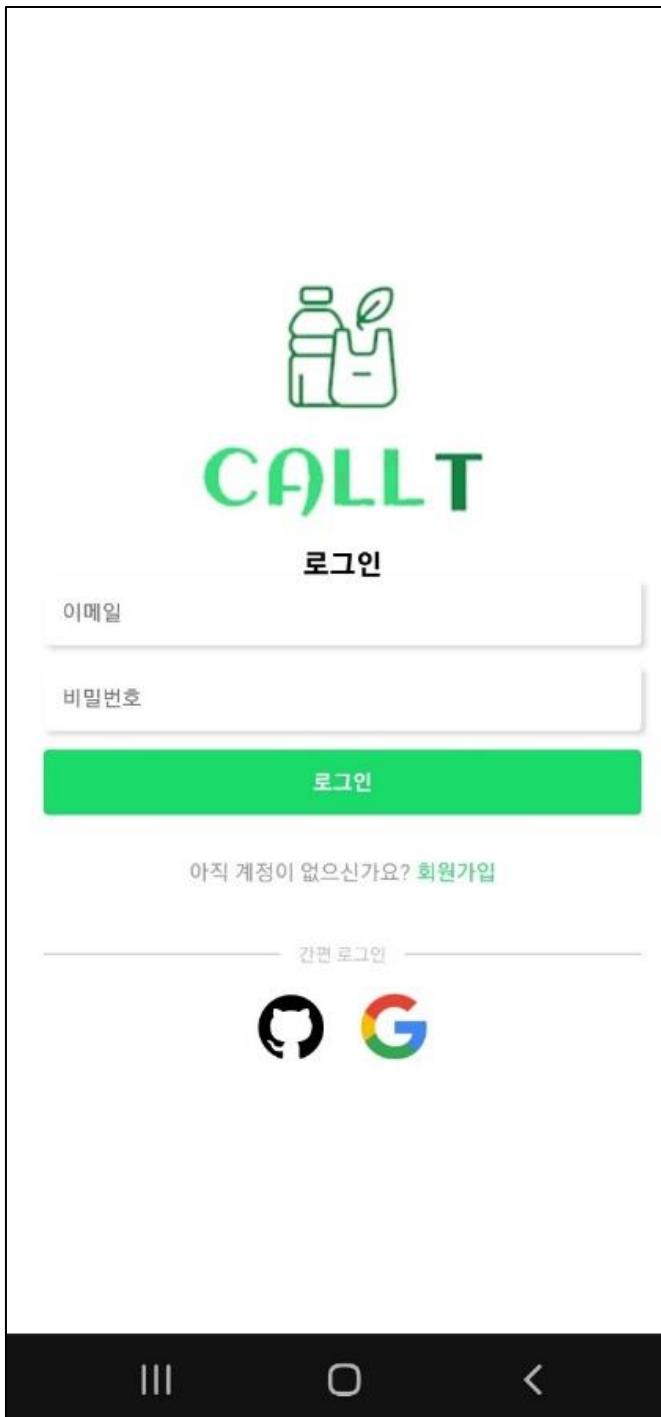
퀴즈 화면

정답 확인 화면

정답 확인 화면

결과 화면

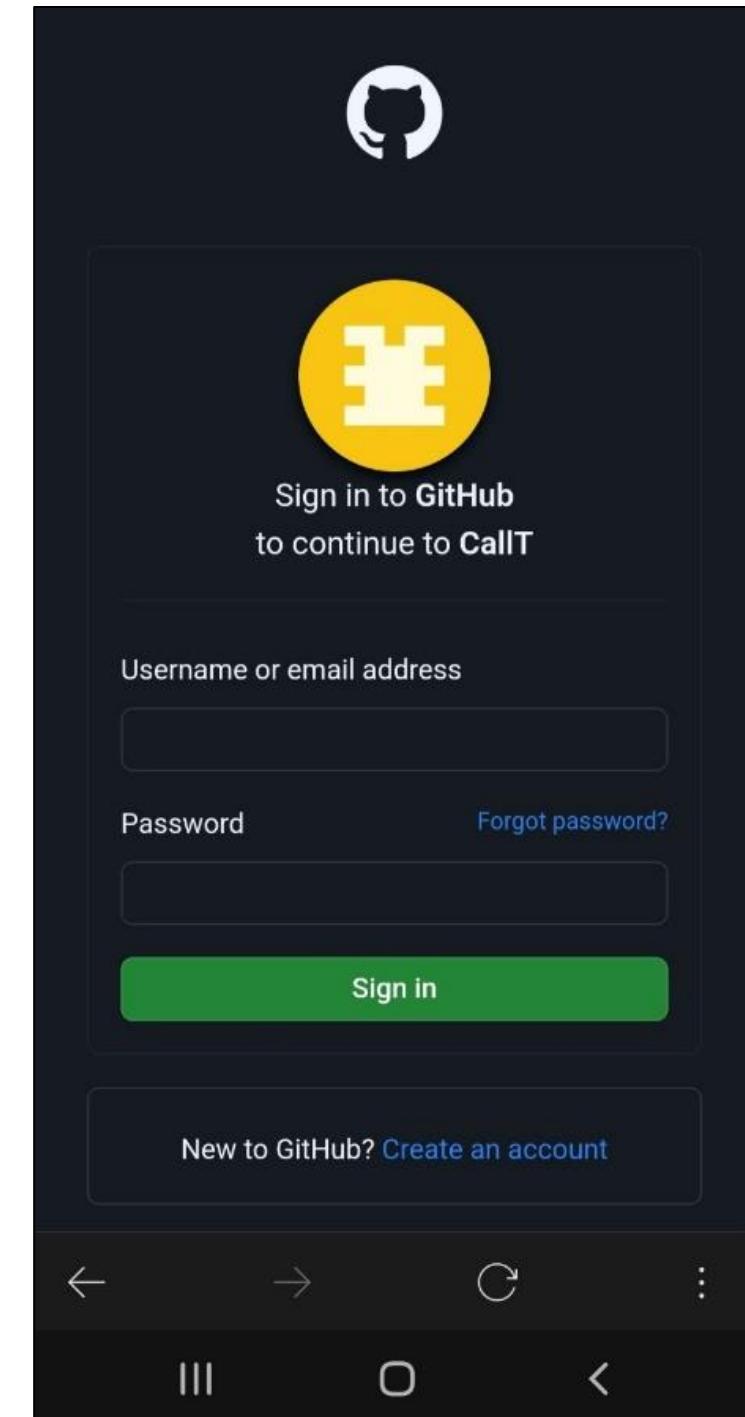
관리자 페이지



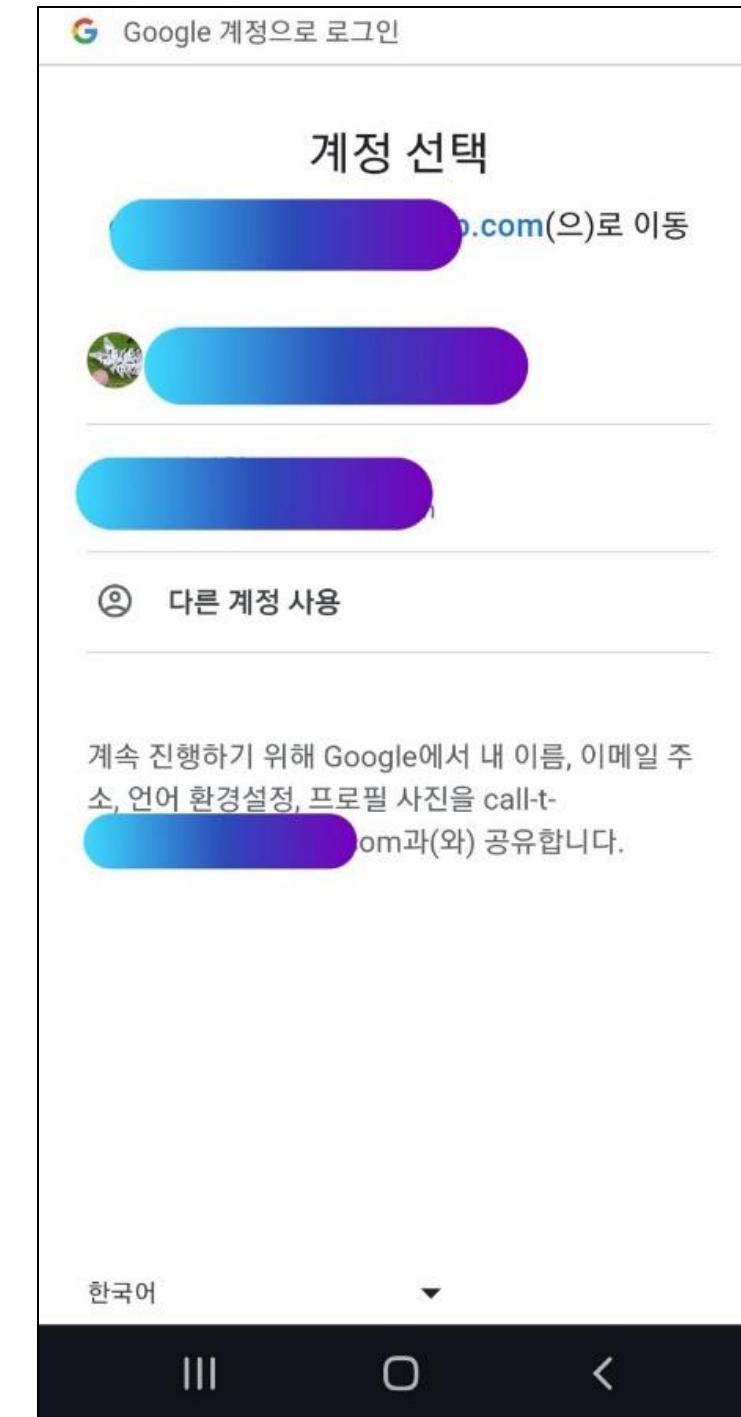
로그인 화면



회원가입 화면

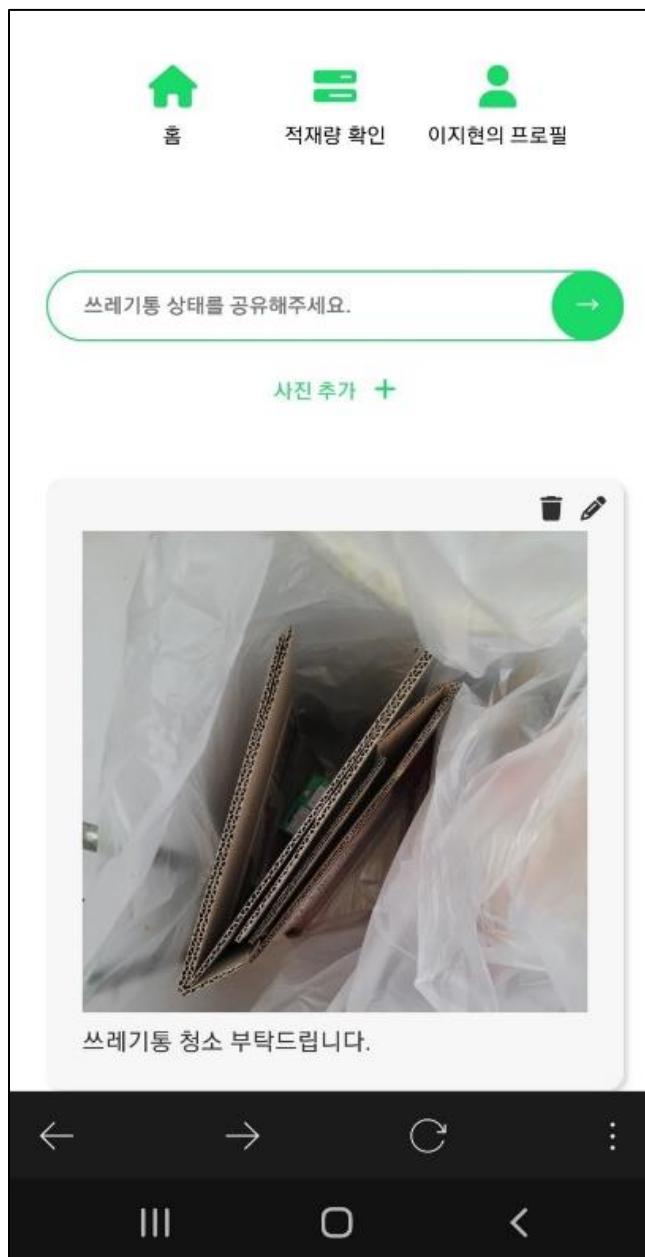


소셜 로그인(깃허브)

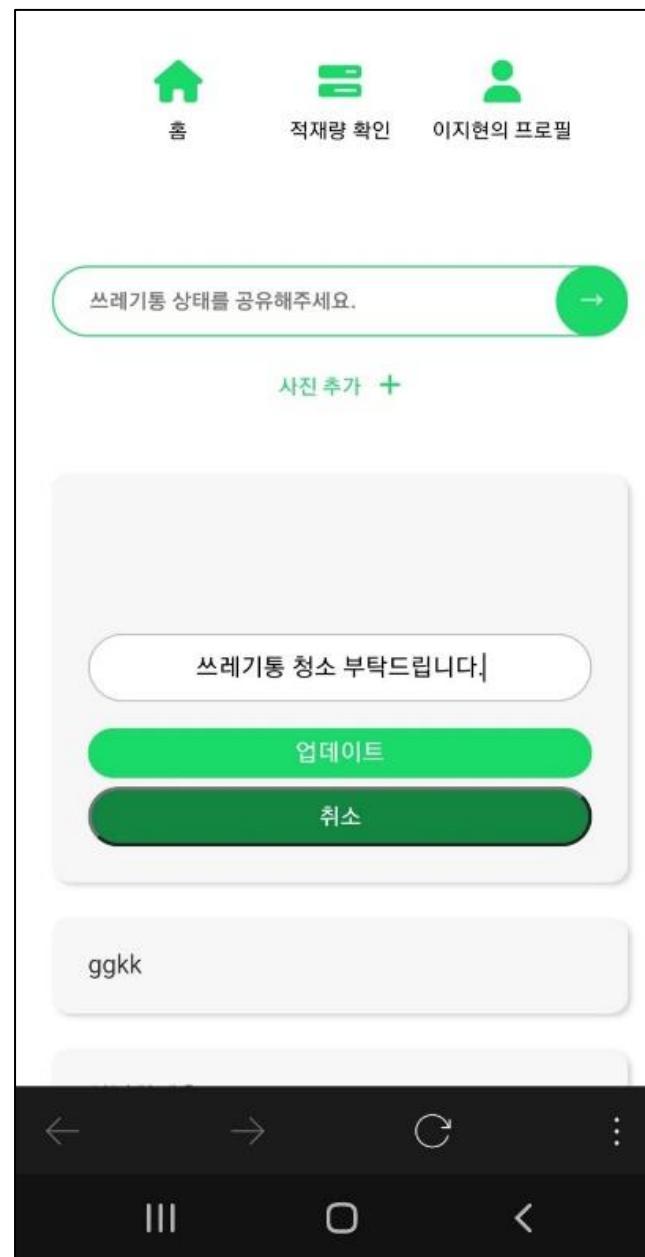


소셜 로그인(구글)

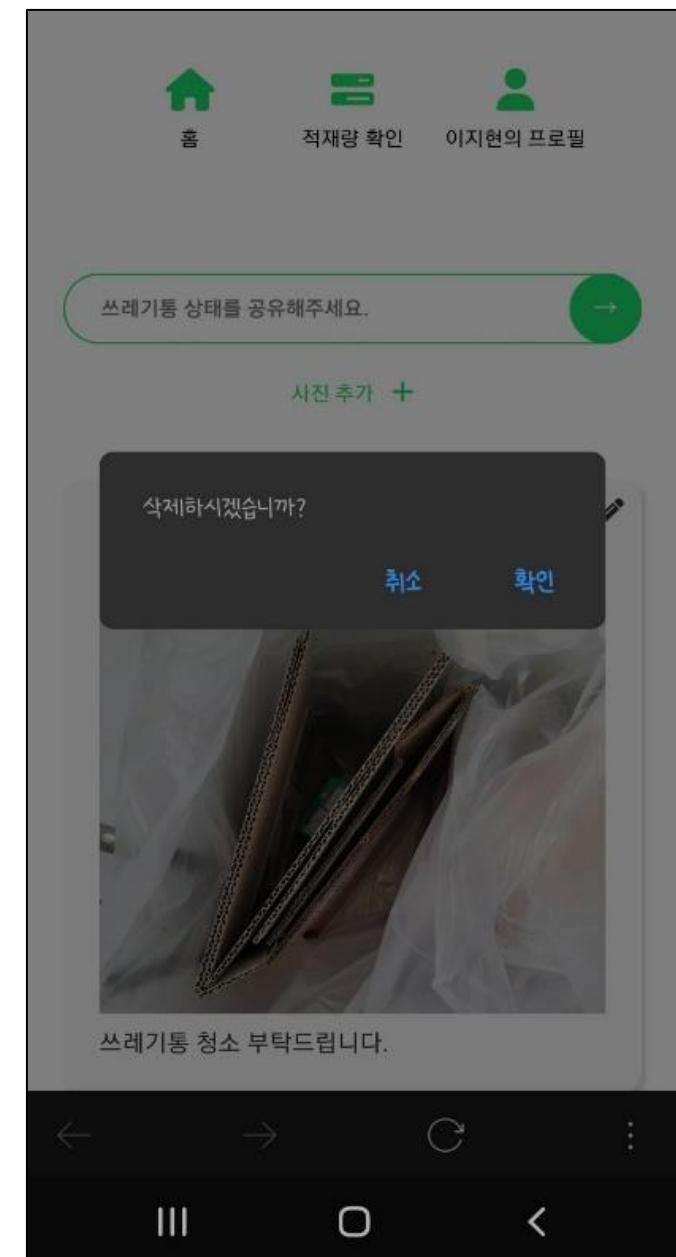
관리자 페이지



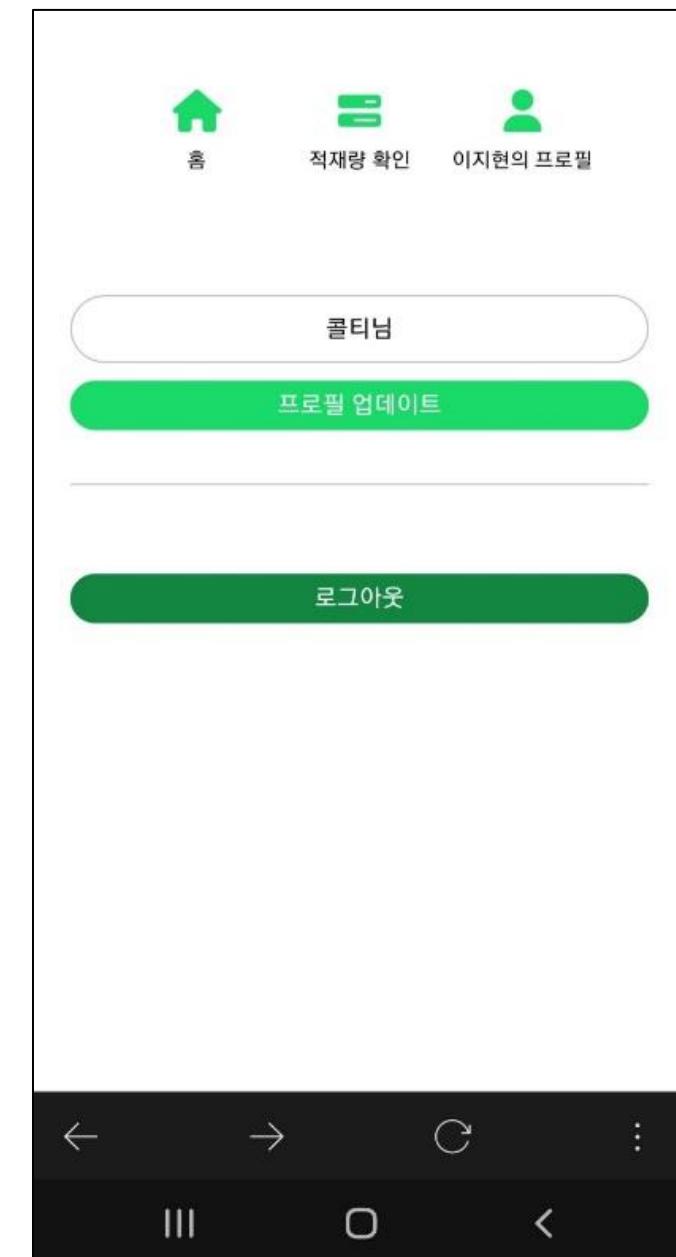
홈 화면



작성 글 업데이트

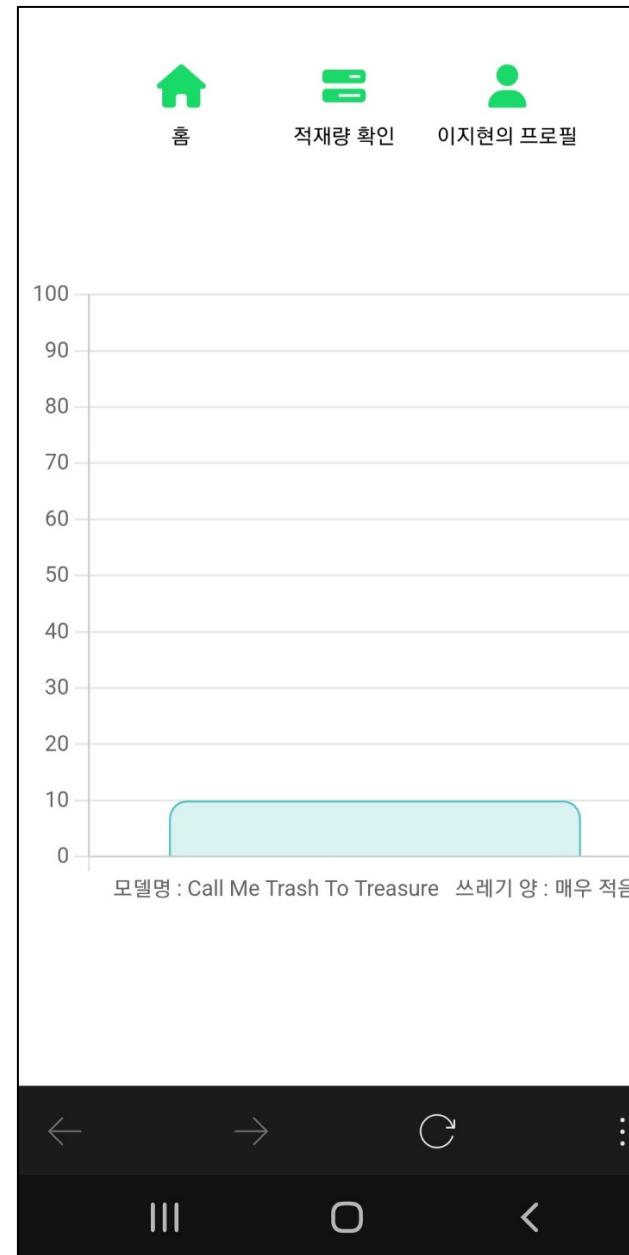


작성 글 삭제



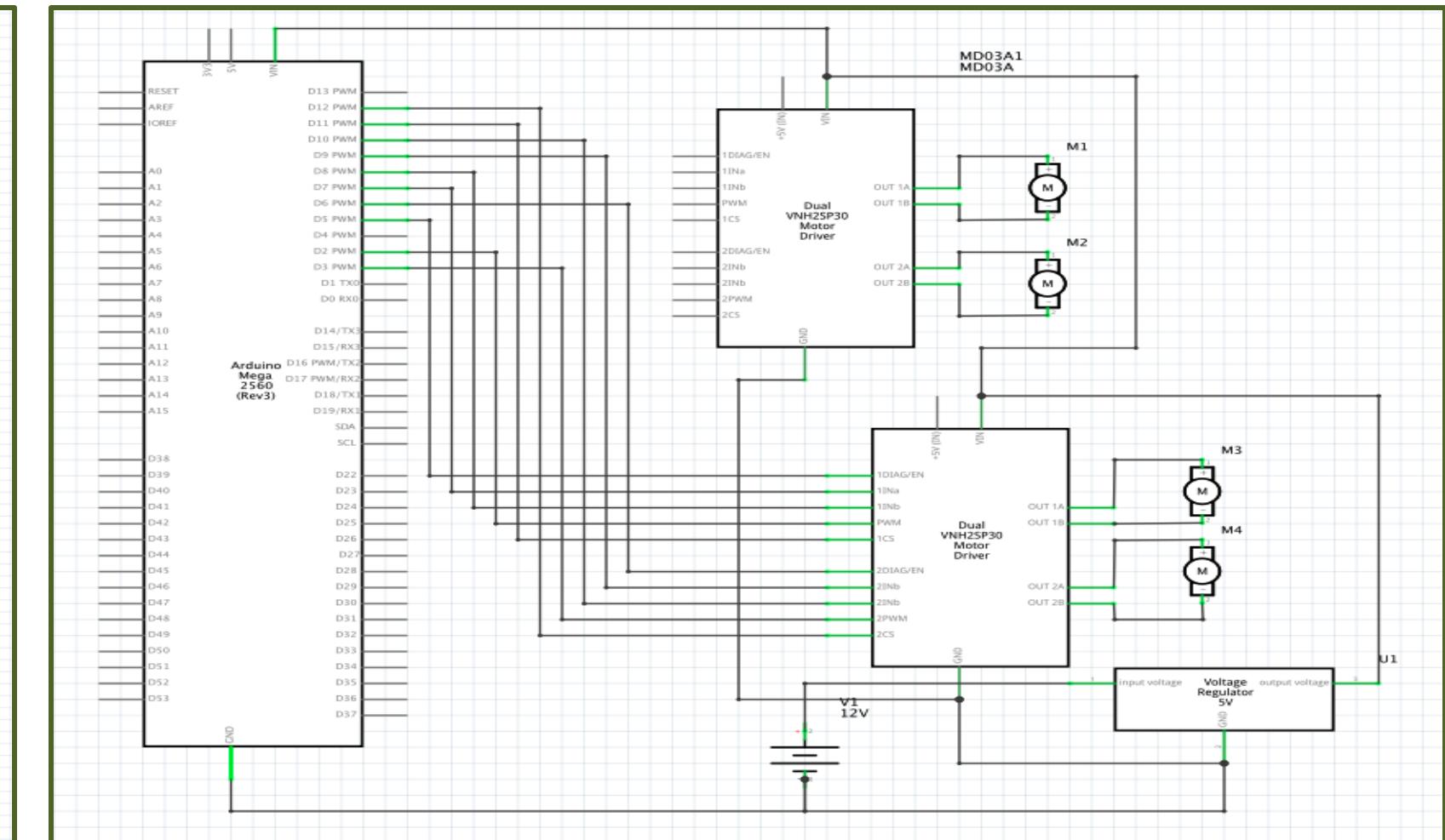
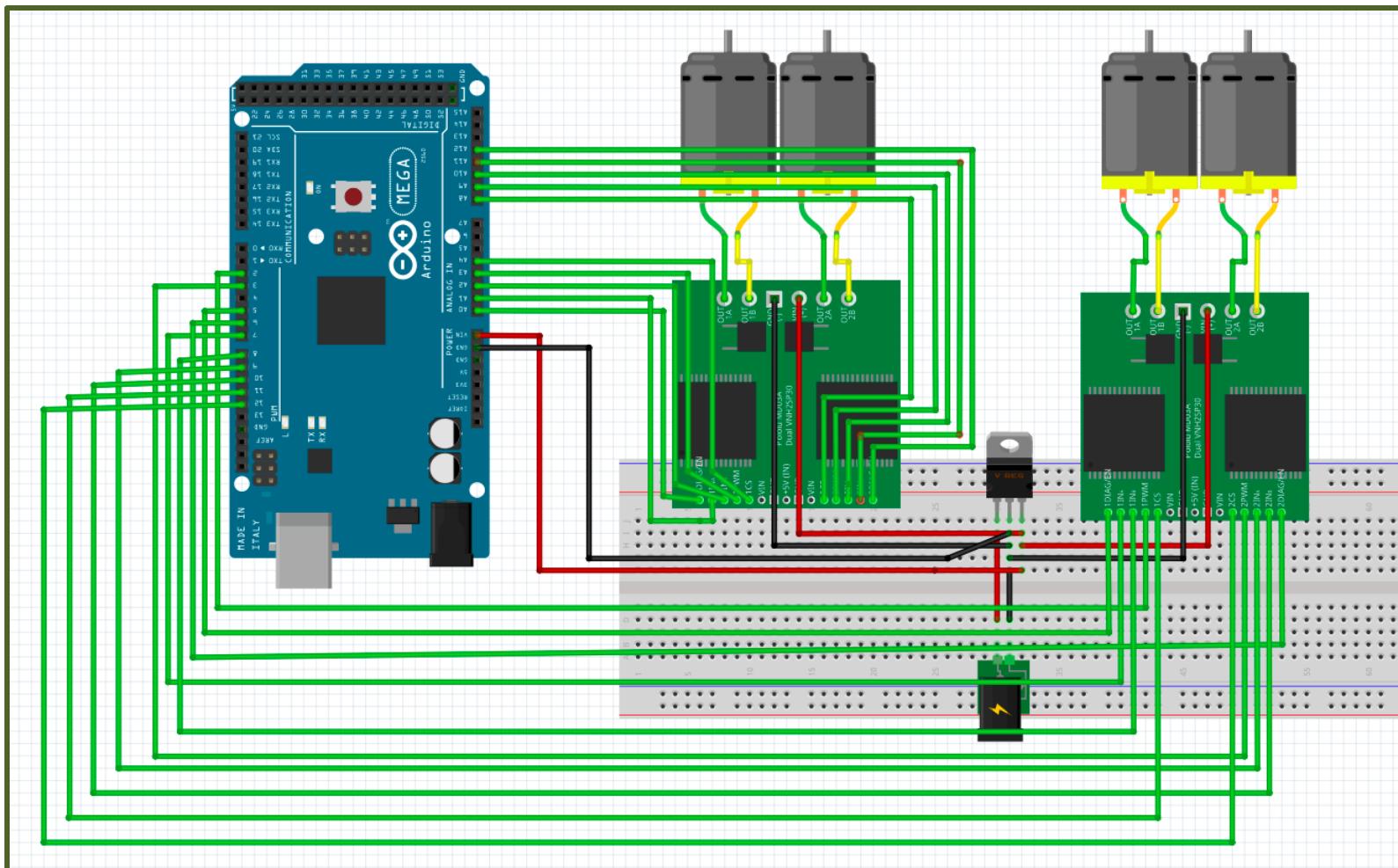
프로필 업데이트

관리자 페이지



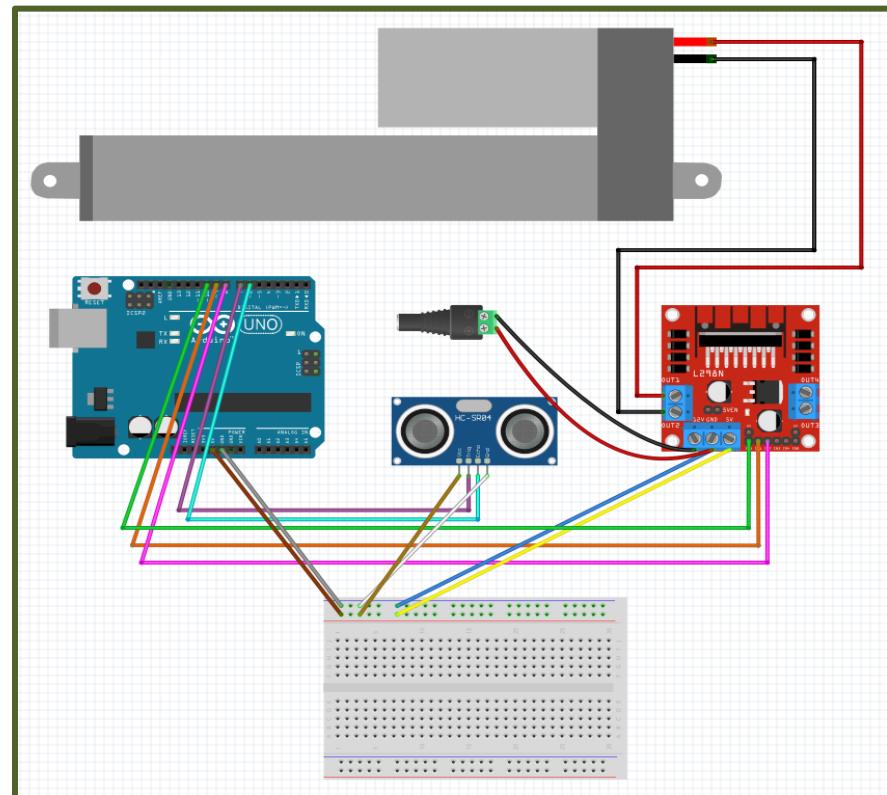
쓰레기적재량 확인

하드웨어/센서 회로도

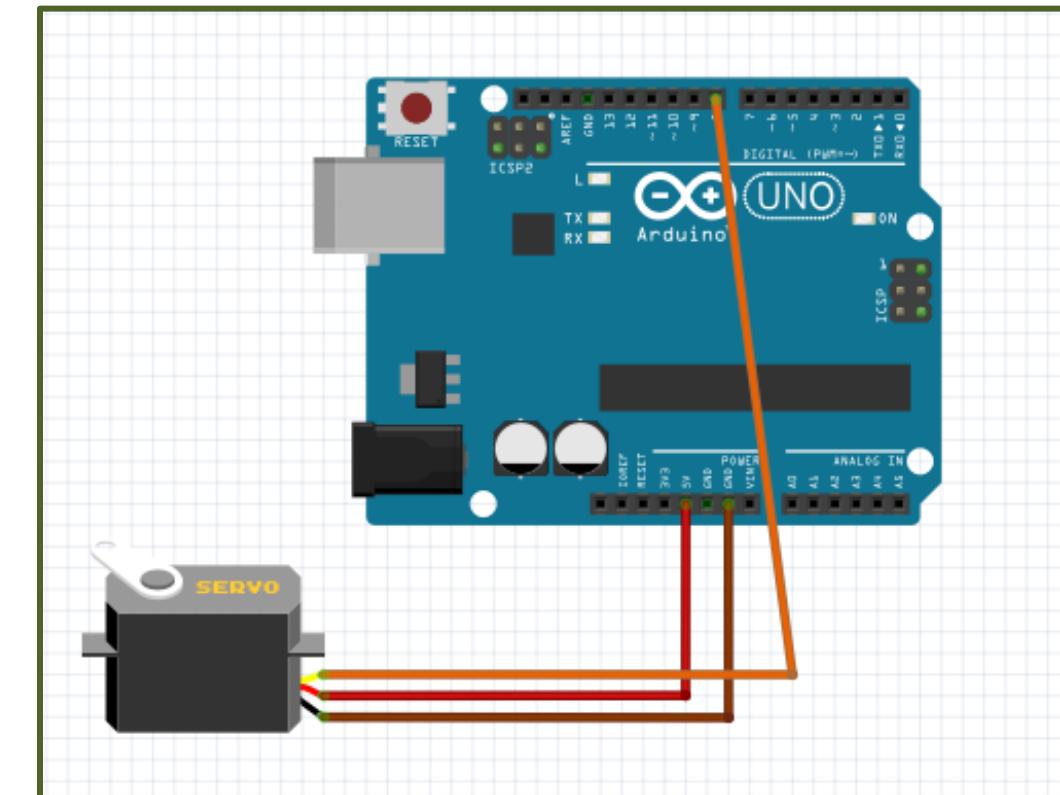


[차체 훨 구성도]

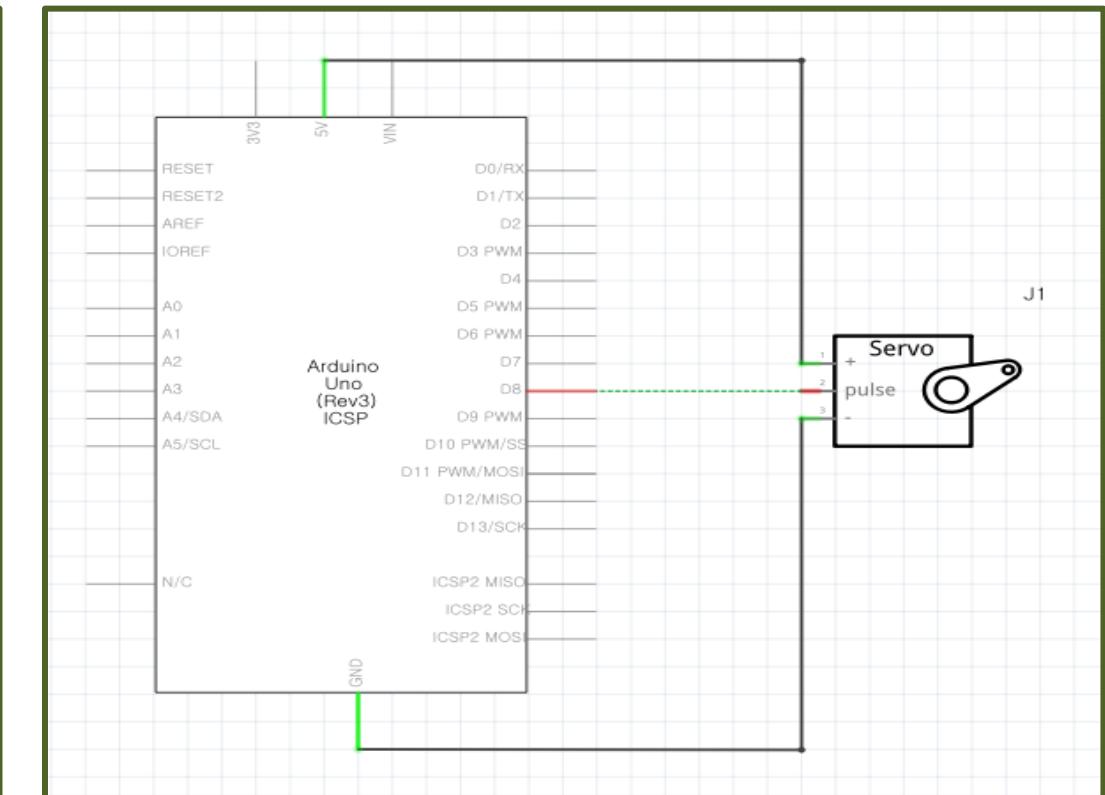
하드웨어/센서 회로도



[압축 설계도]



[쓰레기 투입구 설계도]



S/W 주요기능



호출

- 사용자가 제공받은 URL로 웹사이트에 접속하여 쓰레기통을 호출
- 호출하기 버튼 클릭 시 Geolocation API를 이용해 호출 위치의 위, 경도 값을 수집하여 이동

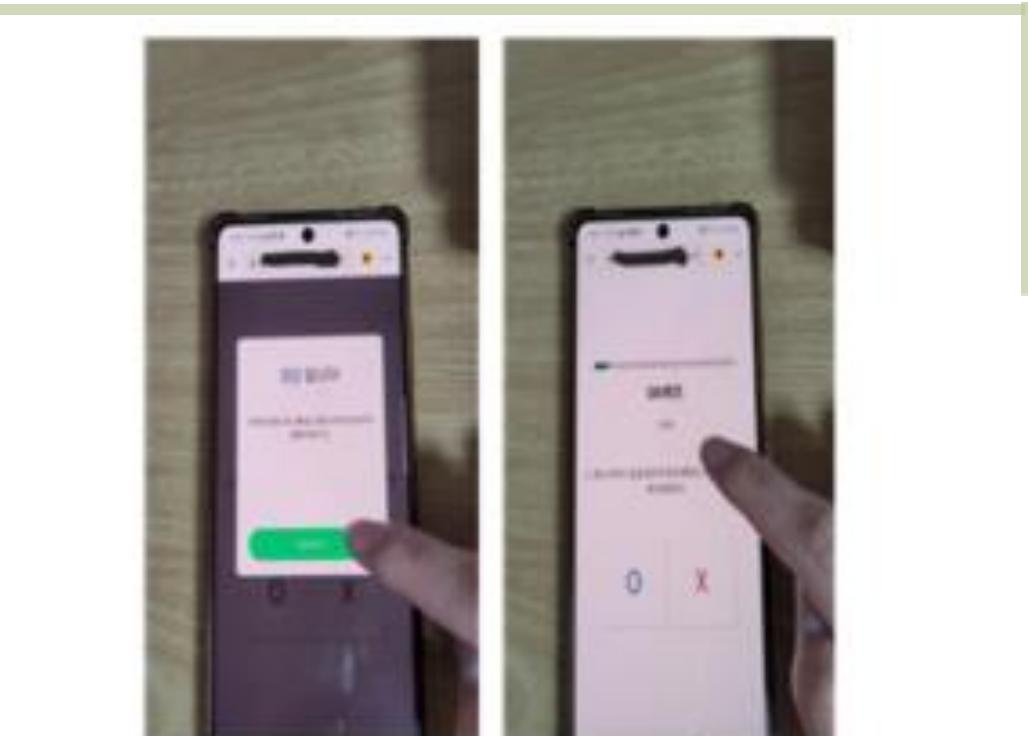
The terminal window shows the following logs:

```
Listening on http://localhost:3001
PS C:\Users\chihg\Desktop\callt\backend> node server.js
listening on http://localhost:3001
PS C:\Users\chihg\Desktop\callt\backend> node server.js
listening on http://localhost:3001
[ 37.18282639999999, 127.12066905000002 ]
[ 37.18282639999999, 127.12066905000002 ]
[ 37.18282639999999, 127.12066905000002 ]
[ 37.18282639999999, 127.12066905000002 ]
[ 37.18282639999999, 127.12066905000002 ]
```

The browser window shows a list of coordinates being collected.

호출 (실외 목적지로 이동 - rest api, geolocation)

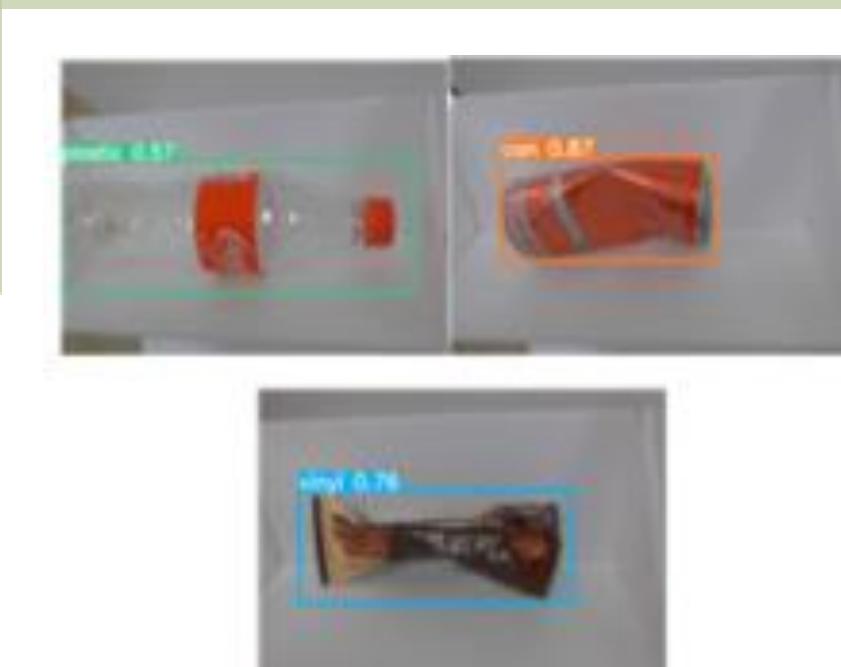
- 사용자가 제공받은 URL에 접속하여 쓰레기통을 호출
- 사용자의 위치 값을 api를 통해 쓰레기통 로봇에 전달



퀴즈

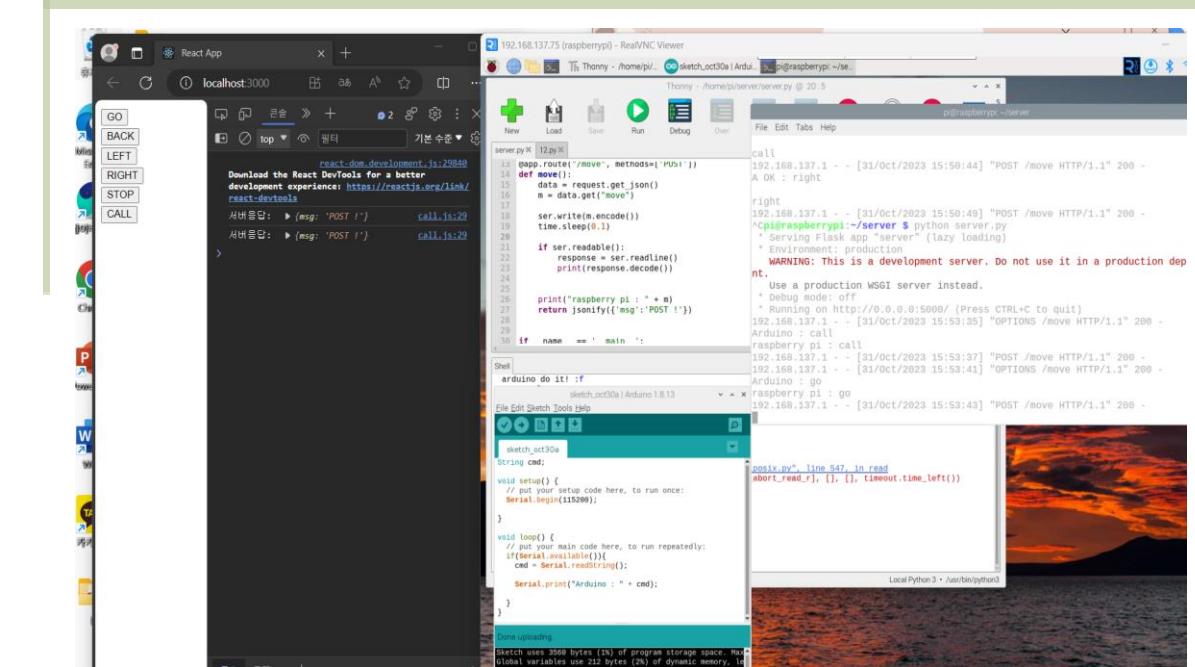
- 쓰레기통을 기다리는 동안 제공된 웹사이트에서 퀴즈 콘텐츠를 진행

S/W 주요기능



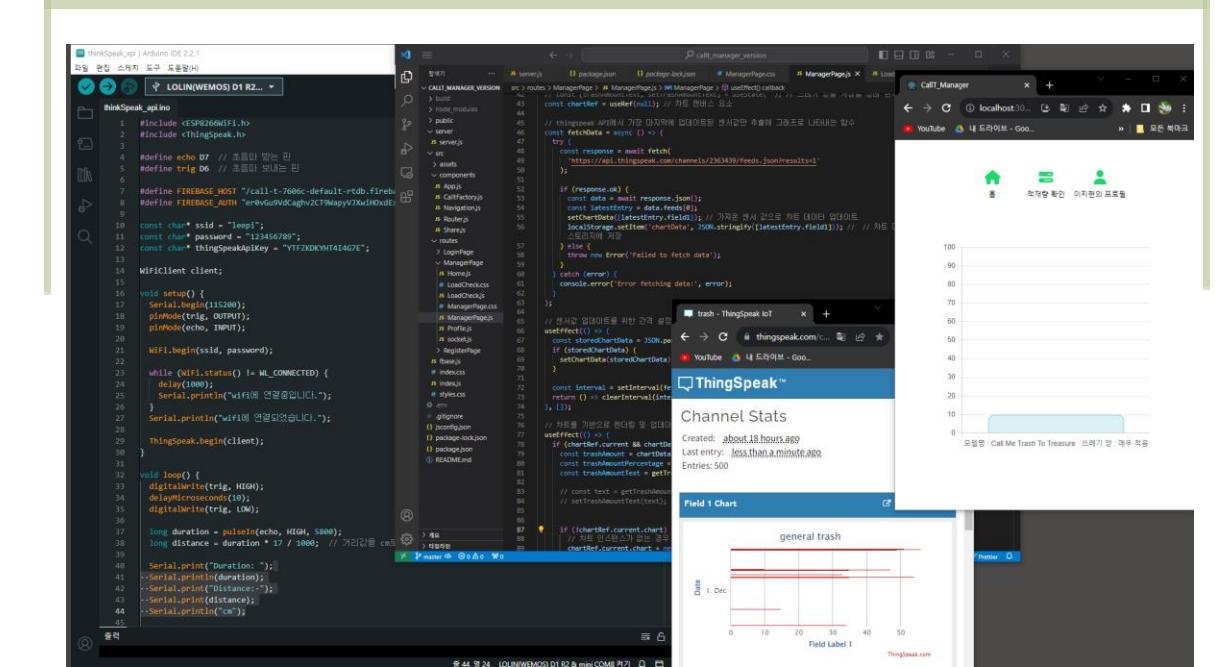
쓰레기 인식

- 커스텀 데이터셋으로 학습시킨 결과 파일을 이용하여 쓰레기통에 투입된 쓰레기를 실시간으로 인식



웹페이지 <-> 라즈베리파이 <-> 아두이노를 연결하는 API 생성 (flask 서버)

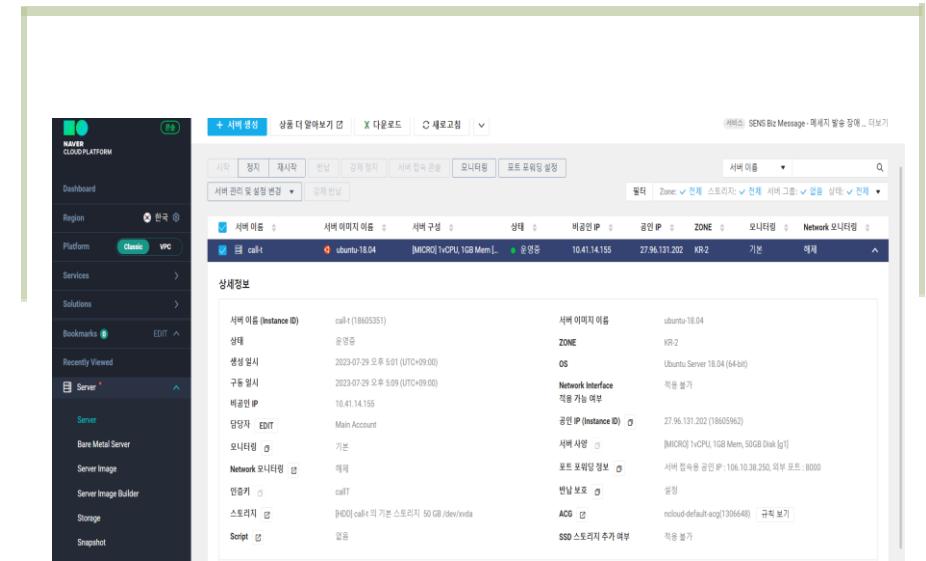
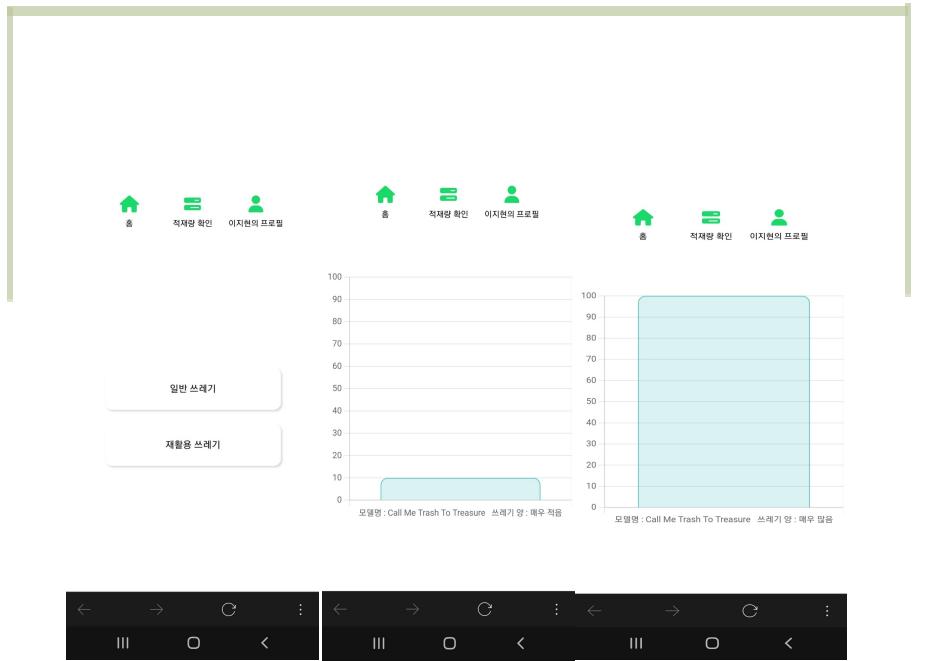
- 사용자가 웹 페이지에서 ‘호출하기’ 버튼을 누르면 라즈베리파이에서 명령을 입력받은 후 아두이노로 전송하는 api를 작성



초음파 센서 <-> 관리자 페이지 연동

- 초음파 센서로 쓰레기 적재량을 측정 후 ThingSpeak API로 전송 -> ThingSpeak API에서 가장 마지막으로 업로드된 센서값을 리액트로 가져와 UI에 표시

S/W 주요기능



관리자 페이지

- 관리자가 쓰레기통 로봇의 상태를 실시간으로 확인할 수 있도록 쓰레기통 적재량을 나타냄.

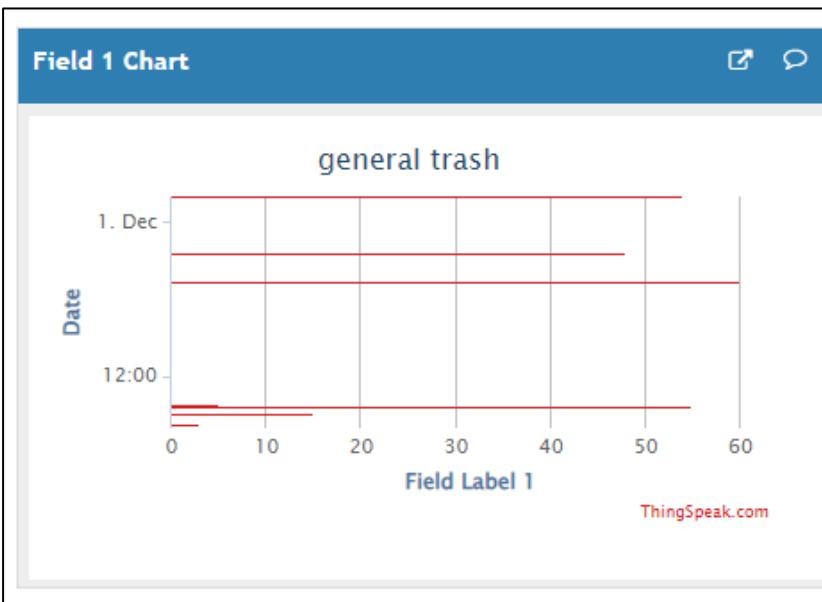
웹 서버 생성

- 네이버 클라우드에서 서버 생성
- python fastapi 이용

S/W 주요기능 쓰레기 적재량 실시간 파악

```
#include <Arduino.h>  
  
#define echo D7 // 초음파 받는 핀  
#define trig D6 // 초음파 보내는 핀  
  
#define FIREBASE_HOST "https://[REDACTED].firebaseio.com/" // firebase 주소  
#define FIREBASE_AUTH "[REDACTED]" // firebase 비밀번호  
  
const char* ssid = "[REDACTED]";  
const char* password = "[REDACTED]";  
const char* thingSpeakApiKey = "[REDACTED]";  
  
WiFiClient client;  
  
void setup() {  
    Serial.begin(115200);  
    pinMode(trig, OUTPUT);  
    pinMode(echo, INPUT);  
  
    WiFi.begin(ssid, password);  
  
    while (WiFi.status() != WL_CONNECTED) {  
        delay(1000);  
        Serial.println("wifi에 연결중입니다.");  
    }  
    Serial.println("wifi에 연결되었습니다.");  
  
    ThingSpeak.begin(client);  
}  
  
void loop() {  
    digitalWrite(trig, HIGH);  
    delayMicroseconds(10);  
    digitalWrite(trig, LOW);  
  
    long duration = pulseIn(echo, HIGH, 5000);  
    long distance = duration * 17 / 1000; // 거리값을 cm로 환산  
  
    Serial.print("duration: ");  
    Serial.println(duration);  
    Serial.print("distance: ");  
    Serial.println(distance);  
    Serial.println("cm");  
  
    ThingSpeak.writeField(2363439, 1, distance, thingSpeakApiKey);  
    delay(2000);  
}
```

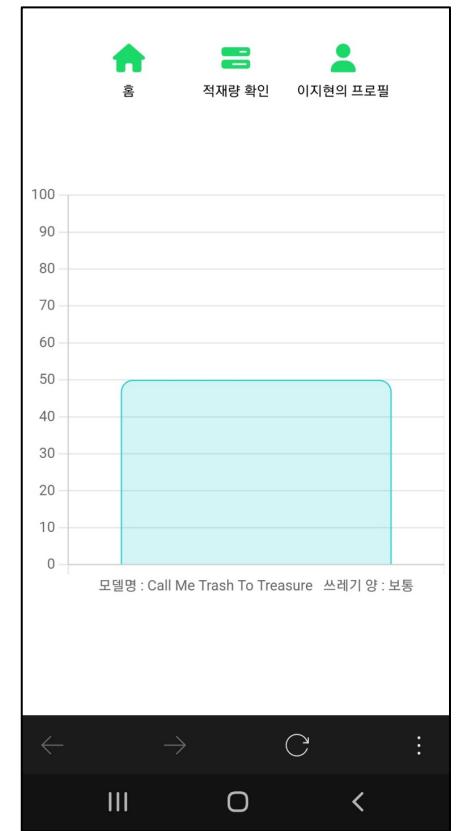
[아두이노 코드에서 센서값 전송]



[ThingSpeak API에서 센서값 수신]

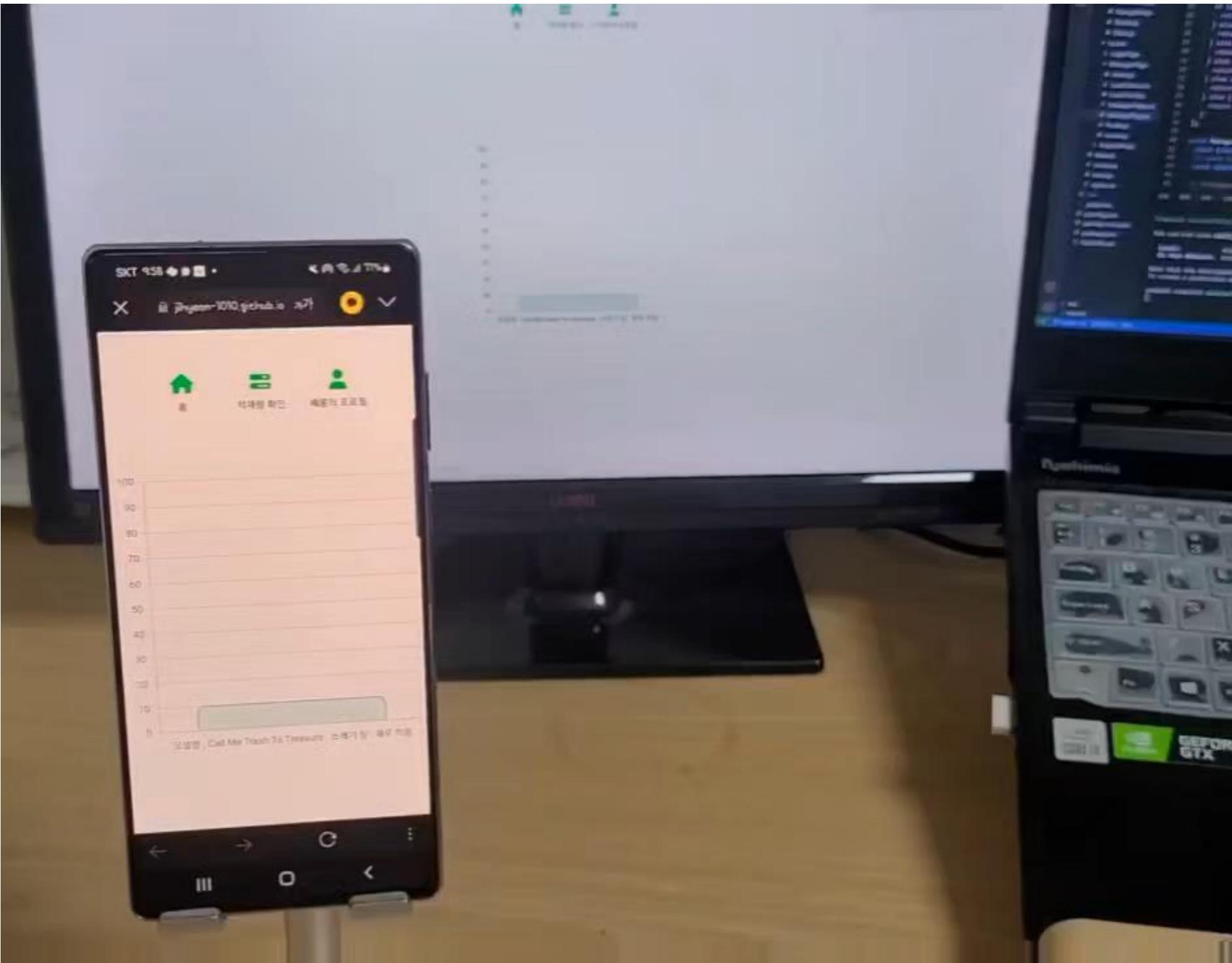
```
Chart.register(Universal, BarController, BarElement, CategoryLabel);  
  
// 쓰레기 적재량 측정하는 법  
const getTrashAmount = (amount) => {  
    if (amount < 10) amount = 10;  
    else if (amount > 50) amount = 50;  
    else if (amount > 10 && amount < 20) return 10;  
    else if (amount > 20 && amount < 30) return 20;  
    else if (amount > 30 && amount < 40) return 30;  
    else if (amount > 40 && amount < 50) return 40;  
    else if (amount > 50 && amount < 60) return 50;  
    else if (amount > 60 && amount < 70) return 60;  
    else if (amount > 70 && amount < 80) return 70;  
    else if (amount > 80 && amount < 90) return 80;  
    else if (amount > 90 && amount < 100) return 90;  
    else if (amount > 100) return 100;  
    else return 0;  
};  
  
// 쓰레기 적재량 보통인 경우  
const getTrashAmountPercentage = (amount) => {  
    if (amount < 10) amount = 10;  
    else if (amount > 50) amount = 50;  
    else if (amount > 10 && amount < 20) return 10;  
    else if (amount > 20 && amount < 30) return 20;  
    else if (amount > 30 && amount < 40) return 30;  
    else if (amount > 40 && amount < 50) return 40;  
    else if (amount > 50 && amount < 60) return 50;  
    else if (amount > 60 && amount < 70) return 60;  
    else if (amount > 70 && amount < 80) return 70;  
    else if (amount > 80 && amount < 90) return 80;  
    else if (amount > 90 && amount < 100) return 90;  
    else if (amount > 100) return 100;  
    else return 0;  
};  
  
// 쓰레기 적재량 대체로  
const TrashLevel = (amount) => {  
    if (amount < 10) return "Underweight";  
    else if (amount > 10 && amount < 20) return "Normal";  
    else if (amount > 20 && amount < 30) return "Overweight";  
    else if (amount > 30 && amount < 40) return "Heavy";  
    else if (amount > 40 && amount < 50) return "Extremely Heavy";  
    else if (amount > 50 && amount < 60) return "Dangerous";  
    else if (amount > 60 && amount < 70) return "Very Dangerous";  
    else if (amount > 70 && amount < 80) return "Extremely Dangerous";  
    else if (amount > 80 && amount < 90) return "Dangerous";  
    else if (amount > 90 && amount < 100) return "Extremely Dangerous";  
    else return "Unknown";  
};  
  
const ThingSpeak = (url) => {  
    const client = new Client(url);  
    const response = client.get();  
    return response;  
};  
  
try {  
    const response = await ThingSpeak("https://api.thingspeak.com/update.json?api_key=2363439&field1=12.00");  
    console.log(response);  
} catch (error) {  
    console.error(error);  
}
```

[리액트에서 센서값 수신]



[관리자 페이지에서 적재량 확인]

S/W 주요기능 쓰레기 적재량 실시간 파악

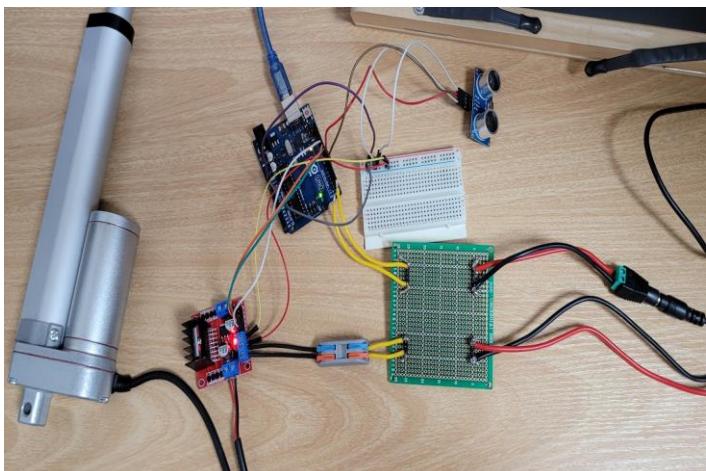


H/W 주요기능 및 부품



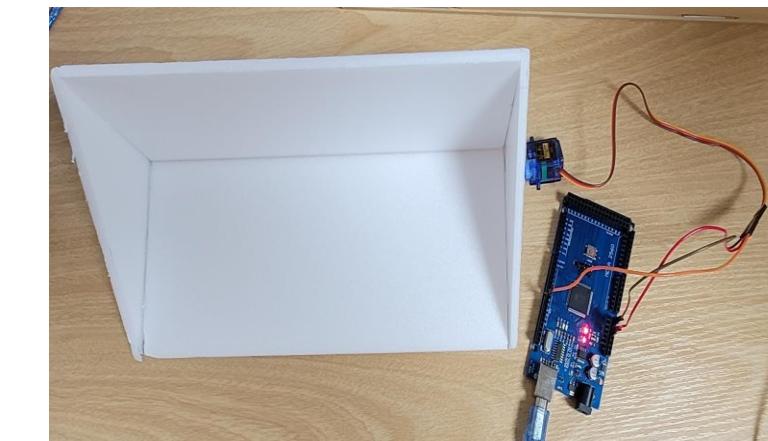
자율주행(아두이노 메가 2560 보드, 4모터드라이버보드, dc 커넥터, 모터 케이블, 초음파 센서, 라이더 센서)

- 목적지까지 장애물을 회피하며 자율주행 (라이더 센서와 초음파 센서를 이용하여 장애물을 인식)



압축(리니어 액추에이터, DC모터 드라이버, 초음파 센서)

- 초음파와 쓰레기 간의 거리 값을 계산해 적재량을 파악한 뒤, 쓰레기가 일정량 이상일 경우 리니어 액추에이터를 구동시켜 압축 진행



쓰레기 분류(서보 모터, 투입구)

- 쓰레기 종류를 인식 후, 재활용 가능 여부에 따라 일반 쓰레기와 재활용 쓰레기로 1차 분류

H/W 주요기능 자율주행

실내 호출

- ESP32-CAM 호환보드를 호출 버튼 형식으로 제작
- 버튼 클릭 시 BLE(Bluetooth Low Energy) 서버가 활성화되어 ESP32 보드의 RSSI(Received Signal Strength Indicator)가 라즈베리파이에 BLE 통신으로 송신
- RSSI를 주기적으로 수신 받는 쓰레기통 로봇은 실내 지도를 기반으로 수신 신호의 세기가 강한 지점으로 이동

$$RSSI = -10n \log_{10}(d) + a$$

(a) RSSI 측정 수식

$$d = 10^{\frac{a-RSSI}{10n}}$$

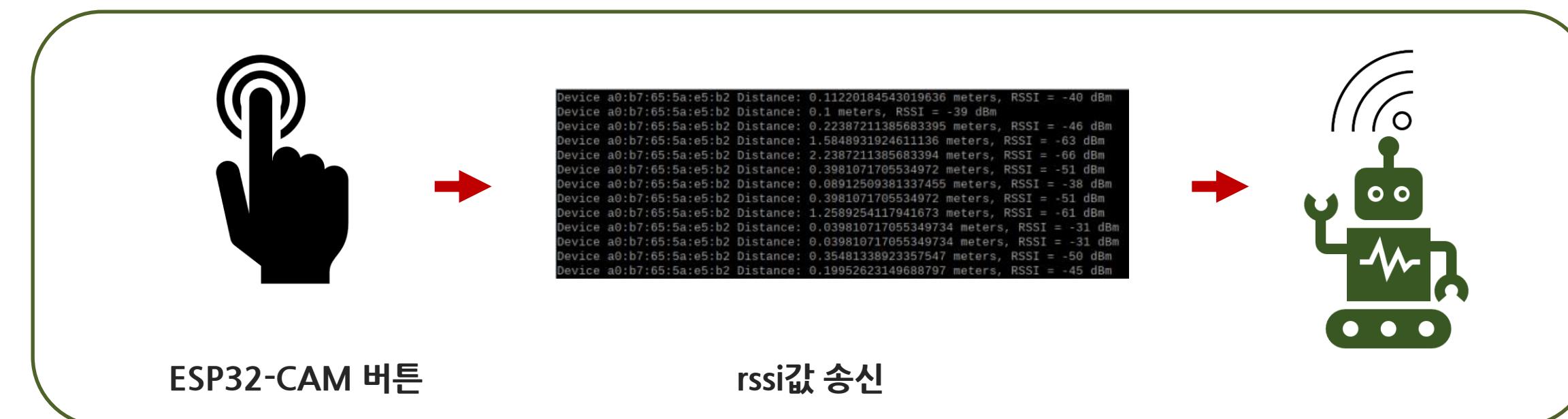
(b) 거리 환산 수식

```
def calculate_distance(rssi):
    tx_power = -59
    n = 2.0

    ratio = rssi * 1.0 / tx_power
    distance = (10 ** ((tx_power - rssi) / (10 * n)))
    return distance
```

```
Device a0:b7:65:5a:e5:b2 Distance: 0.11220184543019636 meters, RSSI = -40 dBm
Device a0:b7:65:5a:e5:b2 Distance: 0.1 meters, RSSI = -39 dBm
Device a0:b7:65:5a:e5:b2 Distance: 0.22387211385683395 meters, RSSI = -46 dBm
Device a0:b7:65:5a:e5:b2 Distance: 1.5848931924611136 meters, RSSI = -63 dBm
Device a0:b7:65:5a:e5:b2 Distance: 2.2387211385683394 meters, RSSI = -66 dBm
Device a0:b7:65:5a:e5:b2 Distance: 0.3981071705534972 meters, RSSI = -51 dBm
Device a0:b7:65:5a:e5:b2 Distance: 0.08912509381337455 meters, RSSI = -38 dBm
Device a0:b7:65:5a:e5:b2 Distance: 0.3981071705534972 meters, RSSI = -51 dBm
Device a0:b7:65:5a:e5:b2 Distance: 1.2589254117941673 meters, RSSI = -61 dBm
Device a0:b7:65:5a:e5:b2 Distance: 0.039810717055349734 meters, RSSI = -31 dBm
Device a0:b7:65:5a:e5:b2 Distance: 0.039810717055349734 meters, RSSI = -31 dBm
Device a0:b7:65:5a:e5:b2 Distance: 0.35481338923357547 meters, RSSI = -50 dBm
Device a0:b7:65:5a:e5:b2 Distance: 0.19952623149688797 meters, RSSI = -45 dBm
```

RSSI와 거리 변환 값



H/W 주요기능 자율주행

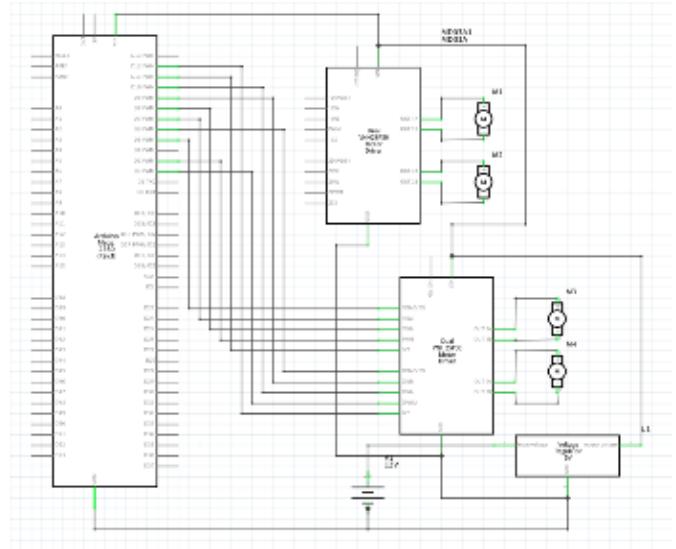
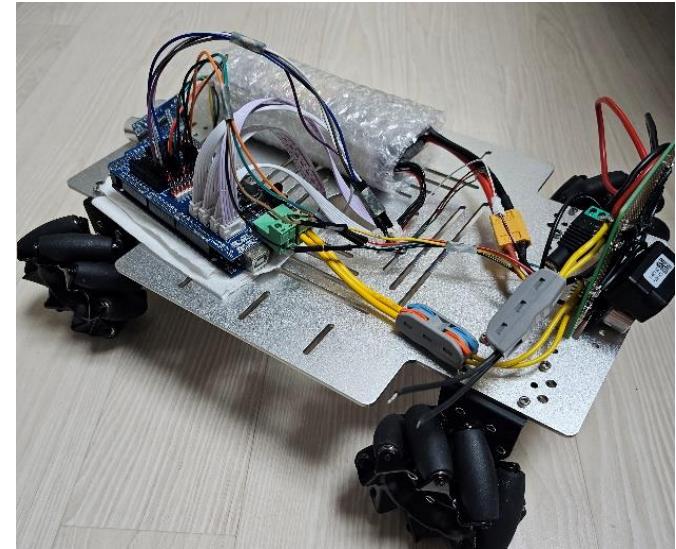


※2배속 영상입니다.

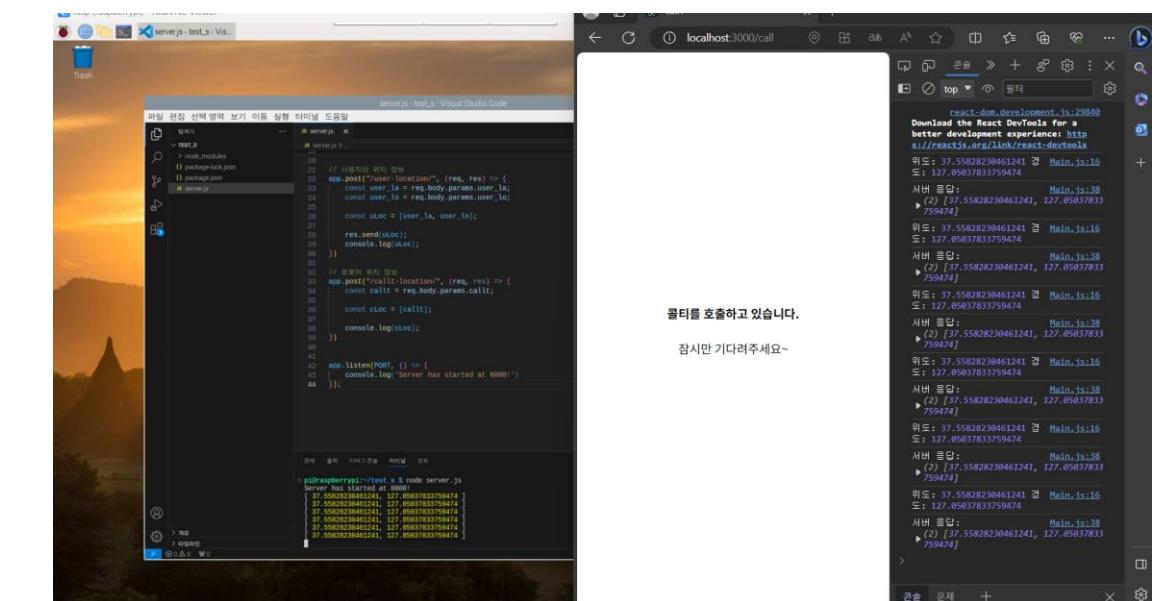
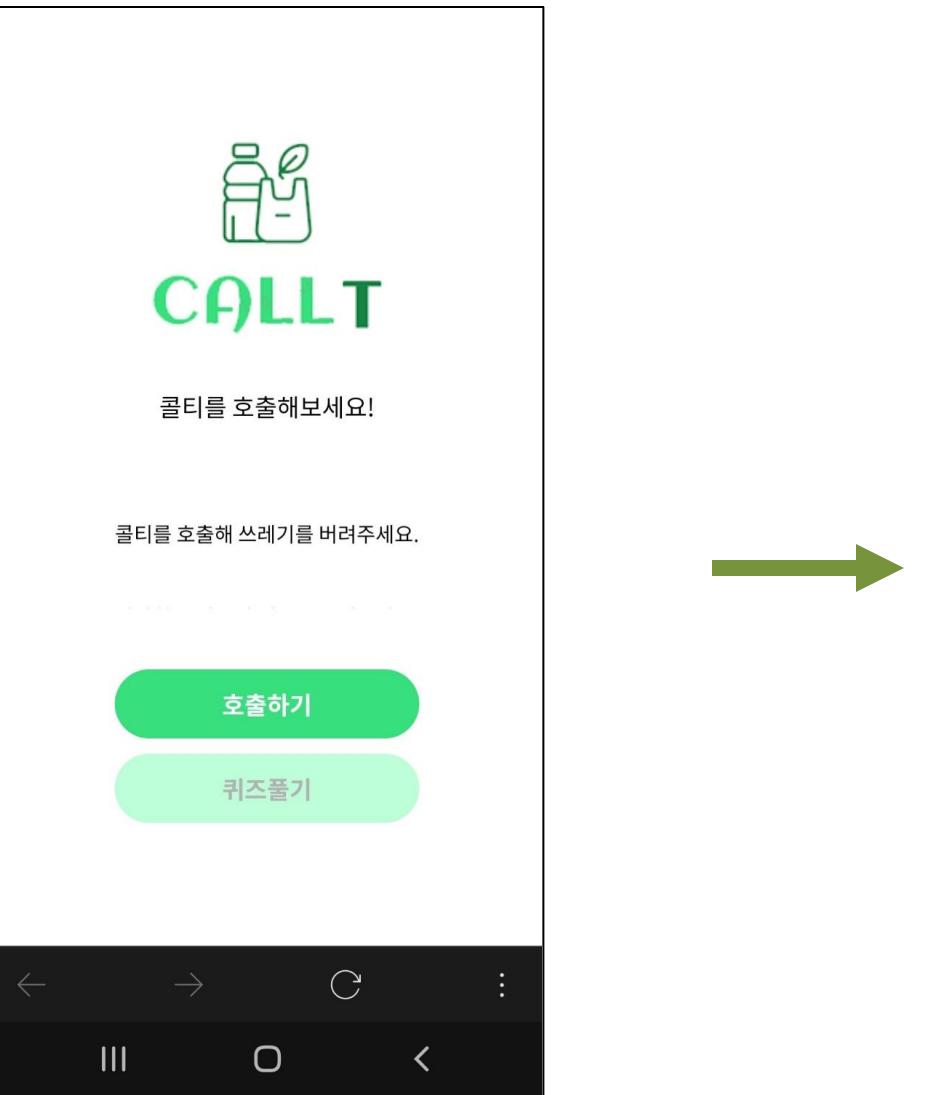
H/W 주요기능 자율주행

실외 호출

- 제공된 URL로 웹사이트에 접속하여 ‘호출하기’ 버튼 클릭
- 로봇은 사용자의 좌표 값을 수집 후 이동



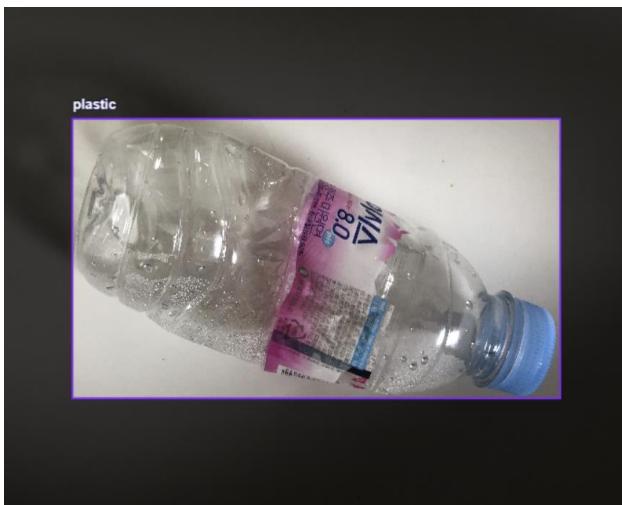
로봇 H/W 구성도



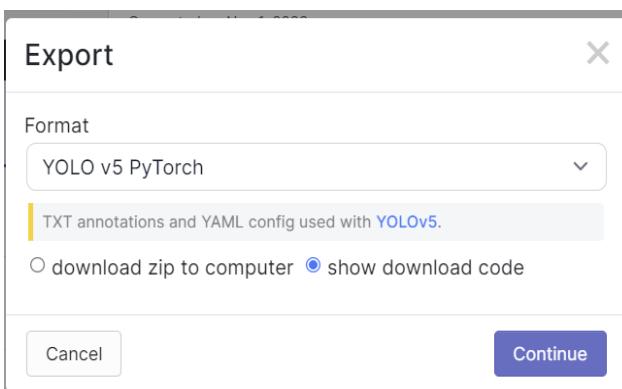
H/W 주요기능 분리수거

- Roboflow를 이용하여 데이터 라벨링
- 생성된 커스텀 데이터셋을 활용하여 YOLOv5 모델 학습
- 학습으로 얻은 weight 파일을 사용하여 카메라에 입력되는 쓰레기를 인식
- 인식된 쓰레기 종류에 따른 서보모터 구동

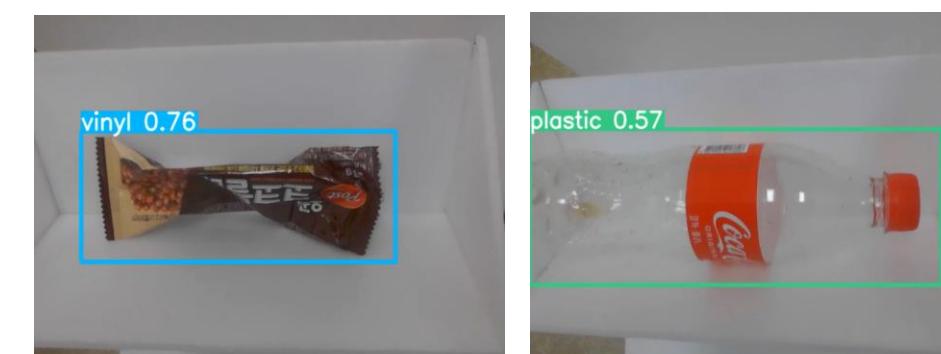
데이터 라벨링



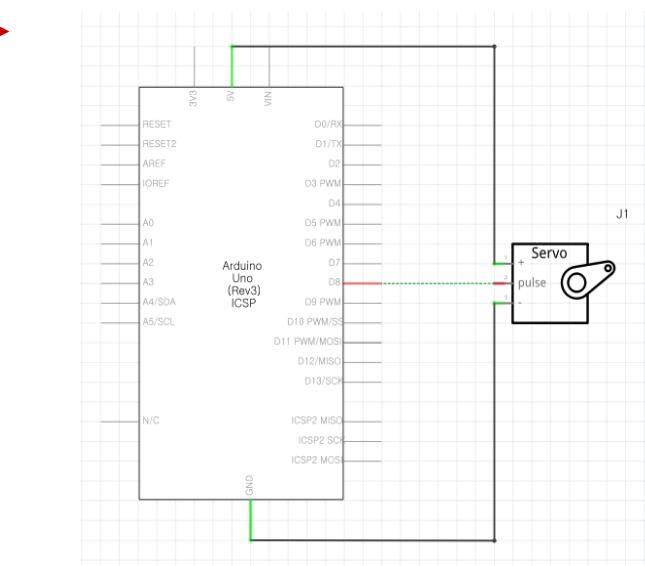
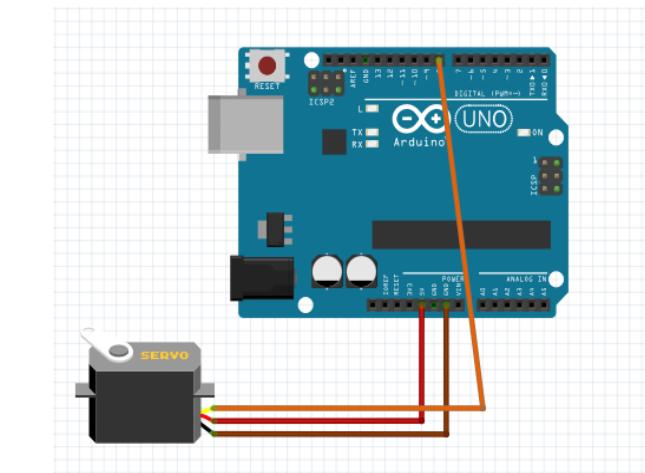
YOLO 모델 학습



카메라를 통해 쓰레기 인식



서보모터 구동



H/W 주요기능 분리수거

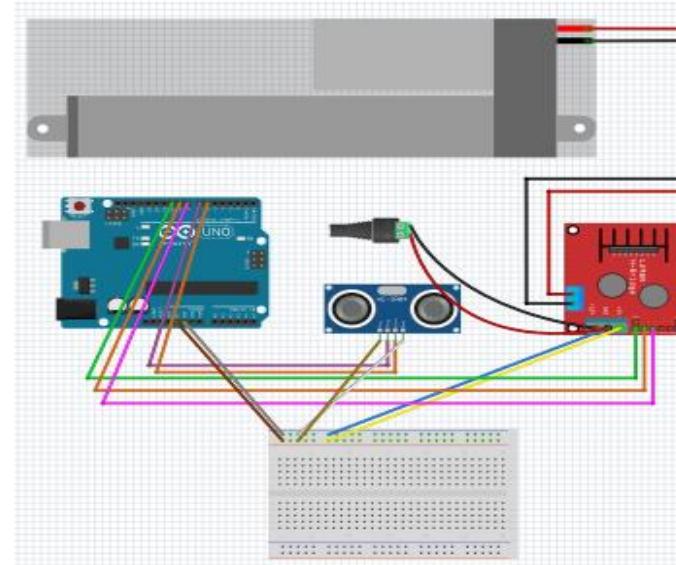
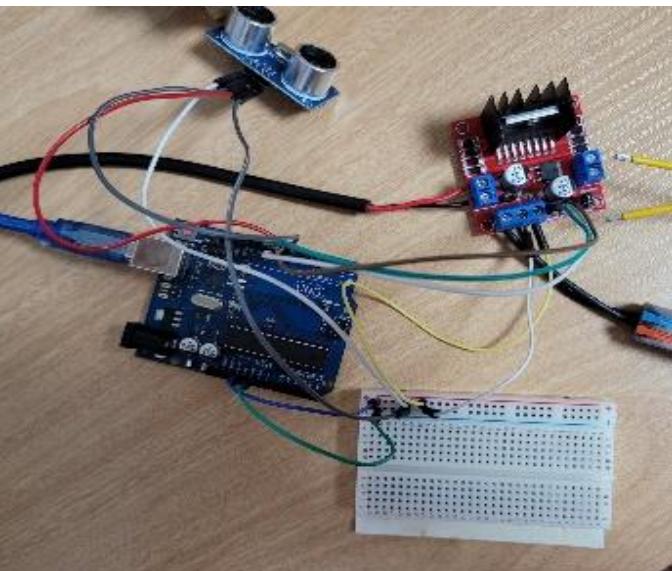


H/W 주요기능 압축

- 쓰레기통 상단에 부착된 초음파 센서 HC-SR04와 적재된 쓰레기 간의 거리를 계산
- 쓰레기가 쓰레기통 높이의 80% 이상 적재 시 자체 압축 진행

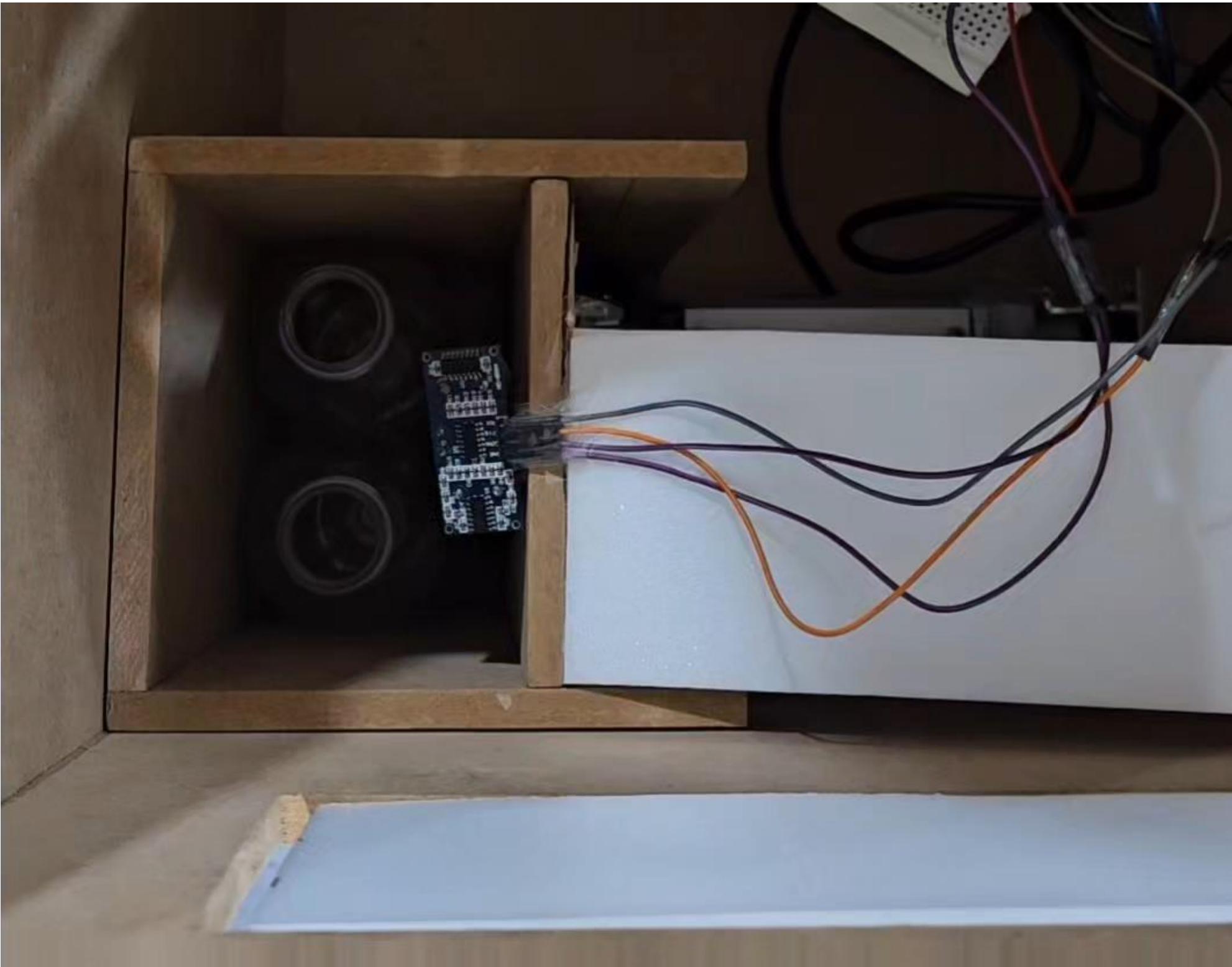
◆ 방법

- NewPing 라이브러리를 사용하여 초음파와 적재된 쓰레기와의 거리를 cm로 환산
- Linear actuator의 양극과 음극 단자를 각각 L298N 모터 드라이버 Output에 연결
- 전원 공급 장치로 12V 5A 어댑터 사용



압축기 H/W 구성도

H/W 주요기능 압축



기술 구현 프로세스

메인 기능

호출 기능

- 1) 호출 버튼을 클릭 시 버튼에 있는 ESP32-CAM 호환보드의 BLE 서버 활성화
- 2) ESP32 보드의 RSSI를 라즈베리파이로 실시간 송신
- 3) 스캔 대기중이던 라즈베리파이에서 ESP32 보드의 RSSI를 실시간 수신
- 4) 시리얼 통신을 통해 라즈베리파이에서 아두이노 메가로 RSSI값을 실시간으로 전송
- 5) 차량은 RSSI값을 비교하며 신호값이 강한 지점으로 이동
- 6) 초음파 센서와 라이더 센서를 이용하여 장애물 회피
- 7) RSSI값이 -35dBm 이상일 때 즉, 호출 버튼 앞에서 정지

압축 기능

- 1) 초음파 센서 HC-SR04를 쓰레기통 상단에 부착한 후 초음파와 적재된 쓰레기 간의 거리를 계산
- 2) NewPing 라이브러리를 사용하여 초음파와 적재된 쓰레기 간의 거리를 cm로 환산
- 3) 10cm 이하일 때 즉, 쓰레기가 쓰레기통 높이의 80% 이상 적재될 시 모터를 구동시켜 리니어 액추에이터 동작
- 4) 쓰레기가 쓰레기통의 90% 이상 적재될 시 firebase에 회원가입 된 관리자에게 이메일 전송

분리수거

- 1) Roboflow를 이용하여 데이터 라벨링 진행
- 2) 라벨링된 이미지로 쓰레기 데이터셋 생성
- 3) Colab 환경에서 YOLOv5 모델을 학습
- 4) 쓰레기가 투입되면 학습으로 얻은 결과 파일을 사용하여 카메라 모듈에 입력되는 쓰레기를 인식
- 5) 쓰레기가 ‘can’, ‘plastic’, ‘paperpack’일 경우 재활용 쓰레기로, ‘paper’, ‘tissue’, ‘vinyl’일 경우 일반 쓰레기로 분류

프로젝트 관리

프로젝트 관리

The screenshot shows a dashboard with several project cards. Each card displays a title, a progress bar, and some descriptive text. Below the cards is a detailed view of a specific project, showing its code repository structure on GitHub. The repository includes files like `index.html`, `index12.png`, `manifest.json`, and `index.html`. The sidebar on the left lists categories such as `Project`, `Task`, `Issue`, and `File`.

형상관리

This screenshot shows a software interface for managing shapes or diagrams. It features a sidebar with navigation links like `Code`, `Issues`, `Pull requests`, `Actions`, `Projects`, `Security`, and `Insights`. The main area displays a tree view of files under a repository named `jhyeon-1010/cabt_user_version`. The tree includes categories like `bin`, `bin/public`, `bin/en`, `bin/img`, and `bin/routes`, each containing various files such as `Address`, `Category`, and `Location`.

의사소통 관리

The screenshot displays a communication management application. At the top, there's a message feed with a blue header that says "안녕하세요. 여러분 ICT면포럼 사용국입니다." Below the feed is a user profile with a picture and the name "jhyeon-1010". The main area shows a list of messages with snippets of text. At the bottom, there's a video conference interface with four participants' video feeds arranged in a grid.

창출하고자 하는 사회적 가치



쓰레기 무단투기 예방

- 쓰레기통 호출 기능을 통해 손쉽게 쓰레기통을 호출하여 쓰레기 무단투기 예방이라는 가치를 창출하고자 함



쓰레기 수거 공간 최대 활용

- 쓰레기가 특정 부하 수준에 도달하면 자체 압축을 진행해 더 많은 쓰레기를 수거할 수 있음.



재활용 비용 절감과 효율성 증대

- 재활용 비용 절감과 효율성 증대: 재활용 가능한 쓰레기만 따로 분류하기 때문에 재활용 프로세스를 촉진하여 폐기물 처리의 환경적 영향을 최소화할 수 있음.

창출하고자 하는 사회적 가치



인적 자원 절약

- 쓰레기 수거 효율성이 증대해 인적 자원을 절약할 수 있음.



‘콜티’ 수요 증대 가능성 향상

- 매년 증가하는 쓰레기량과 환경 정책의 강화로 인해 쓰레기 처리 시장이 성장하고 있어 ‘콜티’의 수요 증대 가능성이 있음



환경 보호에 대한 사람들의 인식 개선

- 단순히 시각적인 홍보용 콘텐츠만 제공하는 것이 아니라, 분리수거 관련 퀴즈 콘텐츠를 제공하여 사람들의 환경 보호에 대한 인식을 개선할 수 있도록 도움을 줌.

창출하고자 하는 사회적 가치



기업의 ESG 경영 실천

- 분리 배출 및 쓰레기 무단투기 예방이 가능하여 환경 생태계를 보호할 수 있으므로 기업은 친환경적인 사회적 가치 창출이 가능해져 ESG 경영을 실현할 수 있음.



산업체의 CSR 활동에 기여

- 친환경적인 쓰레기 관리 및 분리수거를 통해 산업체의 환경 친화적인 이미지를 구축하고, 지속 가능한 경영을 강조할 수 있어 산업체의 CSR 활동에 기여할 수 있음.



기존의 쓰레기통 로봇과 차별화

- 최근 분리수거 교육의 필요성이 대두되면서 분리수거 교육용 콘텐츠로서 활용이 가능하다는 점에서 다른 쓰레기통 로봇과 차별화될 수 있음.